Modular Reconfigurable C4I Interface (MRCI) Phase 1

Test Readiness Review (TRR)

TRR Agenda (1 of 3)

Time	Subject	
0830-0840	**Introductions and Program Status**	**- Park**
0840-0930	MRCI Design Update	
	- MRCI Application Programmers Interface	- Griggs
	- MRCI Common Modules	- Hieb/ Silva
	- MRCI Run Time Infrastructure Interface	-McKenzie
0930-0950	CTAPS Update	- Bretton
	- SSI Implementation	
	- Mission Threads / Messages	- Ashley
0950-1000	Break	

TRR Agenda (2 of 3)

Time	Subject	
1000-1020	MCS/P Update	- Howard
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1020-1040	AFATDS Update	- Anglin
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1040-1100	Simulation Federate Update	- Hieb
1100-1120	Test Program	- Chen
	- CT-5	
	- Post February MRCI Assessment Opportunities	
	- CBS JTC Update	

TRR Agenda (3 of 3)

Time	Subject	
1120-1150	MRCI Demonstrations	- Griggs/ Hieb
1150-1200	Wrap-up	- Park
1200	Adjourn	

Program Activity Since PTRR

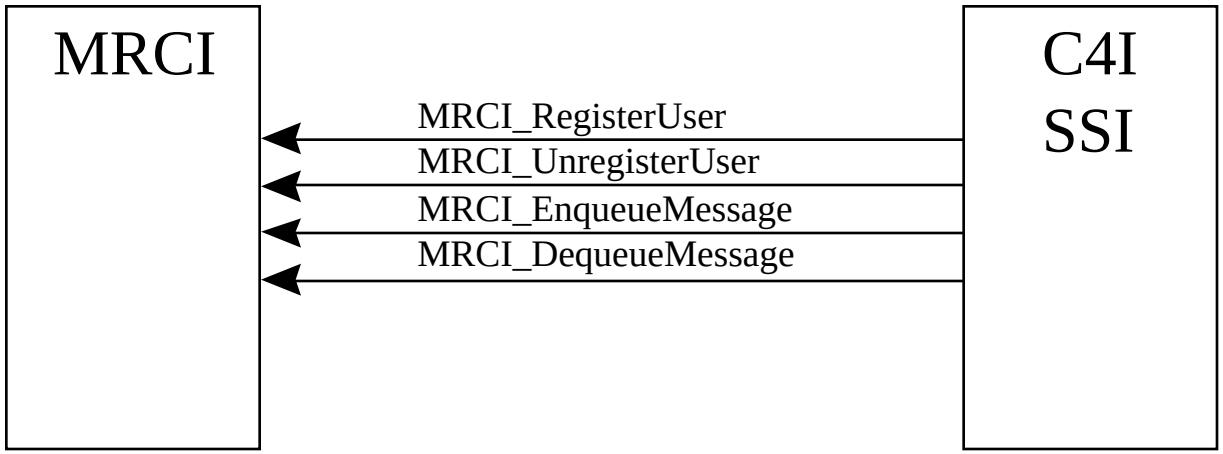
- **Emphasis on enhancing code due to issues discovered during CT-4**
- **Updating the MRCI build schedule**
- **Meetings:**
  - **Univ. of Michigan aWOC Coordination**
  - **WPC Germany, CBS MRCI interface**

TRR Agenda (1 of 3)

Time	Subject	
0830-0840	Introductions and Program Status	- Park
0840-0930	MRCI Design Update	
	- MRCI Application Programmers Interface	- Griggs
	- MRCI Common Modules	- Hieb/ Silva
	- MRCI Run Time Infrastructure Interface	-McKenzie
0930-0950	CTAPS Update	- Bretton
	- SSI Implementation	
	- Mission Threads / Messages	- Ashley
0950-1000	Break	

MRCI
Application Programmers Interface (API)

MRCI API Functions



MRCI API Data Structures (1 of 3)

1.0 Data Structures

```
typedef enum {
        MRCI_SUCCESS = 0,
        MRCI_FAIL_MSG_REPLICATE,
        MRCI_FAIL_INVALID_DATA_TYPE,
        MRCI_FAIL_INVALID_MSG_TYPE,
        MRCI_FAIL_NO_MSG_ID,
        MRCI_FAIL_INVALID_PROT,
        MRCI_FAIL_INV_DATA_FILE,
        MRCI_FAIL_INV_LOG_FILE,
        MRCI_FAIL_INV_CONFIG_FILE,
        MRCI_FAIL_UNKNOWN_PROT,
        MRCI_FAIL_EXCESS_DATA,
        MRCI_FAIL_BUFFER_OVERFLOW,
        MRCI_FAIL_EMPTY_Q,
        MRCI_FAIL_REGISTERUSER,
        MRCI_FAIL_SERVICE,
        MRCI_FAIL_USERIDINVALID,
        MRCI_FAIL_NULLMESSAGE,
        MRCI_FAIL_USERINFOINVALID,
        MRCI_FAIL_NO_LOCAL_USER,
        MRCI_FAIL_NONUNIQUEUSERID,
        MRCI_FAIL_INV_SYSTEM_TYPE,
        MRCI_FAIL_CFG_SYSTEM_TYPE,
        MRCI_FAIL_OUT_OF_MEMORY
} MRCI_Status;
```

MRCI API Data Structures (2 of 3)

```
typedef enum {
        MRCI_OTHER,  /* by convention OTHER is 0 */
        MRCI_AFATDS,
        MRCI_MCSPBL,
        MRCI_CTAPS,
        MRCI_ARSAF,
        MRCI_AFSAF,
        MRCI_EAGLE
} MRCI_SystemType;

typedef enum {
        MRCI_USMTF,
        MRCI_ATCCS,
        MRCI_TACFIRE,
        MRCI_CCSIL,
        MRCI_OTHER_FORMAT
} MRCI_MessageFormatType;

typedef enum
{
        NO_COMP,
        MODSAF_COMP
} MRCI_CompressionType;

typedef enum
{
        NO_ENCAP,           /* no_encapsulation */
        CCSIL_ONLY,         /* ccsil header encapsulation only */
        CCSIL_SIGNAL,       /* ccsil header then signal PDU */
        JANAP128,           /* JANAP128 message header and foot */
        ULP_HEADER          /* Upper Level Protocol header */
} MRCI_EncapsulationType;
```

MRCI API Data Structures (3 of 3)

```
typedef struct m_user_attributes {
  char                    user_name[9];       /* user login name */
  MRCI_SystemType         system_type;
  char                    user_call_sign[40]; /* user's tactical call sign */
  char                    user_role[40];      /* user's tac role (e.g., S3_25INT) */
  char                    user_address[70];   /* hostname or IP address */
  MRCI_EncapsulationType  message_encap;   /* C4I system encapsulation method */
  MRCI_CompressionType    message_comp;    /* C4I system compression Method */
} MRCI_UserAttributes;
```

MRCI API Functions (1 of 5)

Function Parameter Passing Conventions

Parameter Code	Explanation
P1	In parameter by value
P2	In parameter by reference. Caller provides memory. Caller may free memory or overwrite it upon completion of the call. Callee must copy during call anything it wishes to save beyond completion of the call.
P3	Function return by value
P4	Out parameter by reference. Caller provides memory. Caller may free memory or overwrite it upon completion of the call. Callee must copy during call anything it wishes to save beyond completion of the call

MRCI API Functions (2 of 5)

Function Name: **MRCI_RegisterUser**

Purpose:
This function is provided to register a C4I system with the MRCI system.

Calling Sequence:
MRCI_Status status = MRCI_RegisterUser (system, (int *) &user_id)

Parameters

Name	Code	Type	Description
system	P1	MRCI_UserAttributes	Connecting system's attributes
user_id	P4	int *	Pointer to location for MRCI to store ret value

MRCI API Functions (3 of 5)

Function Name: **MRCI_UnregisterUser**

Purpose:
This function is provided to un-register a C4I system with the MRCI system.

Calling Sequence:
MRCI_Status status = MRCI_UnRegisterUser (user_id)

Parameters

Name	Code	Type	Description
user_id	P1	int	MRCI assigned user id returned from MRCI_RegisterUser

MRCI API Functions (4 of 5)

Function Name : **MRCI_EnqueueMessage**

Purpose: This function is provided to allow the C4I system to send messages to MRCI.

Calling Sequence

MRCI_Status status = MRCI_EnqueueMessage(user_id, format, (void *) &message)

Parameters

Name	Code	Type	Description
user_id	P1	int	MRCI assigned user id returned from MRCI_RegisterUser
format	P1	MRCI_MessageFormatType	Format of message (e.g., TACFIRE)
message	P2	void **	Pointer to message
msg_size	P1	int	Size of the message to be enqueued

MRCI API Functions (5 of 5)

Function Name : **MRCI_DequeueMessage**

Purpose:
This function is provided for C4I system to remove a message from the MRCI queue, typically to read the message. MRCI will return the highest priority message on the queue. If there are no messages for the calling C4I system, then MRCI will return an MRCI_FAIL_EMPTY_Q status.

Calling Sequence :
MRCI_Status status = MRCI_DequeueMessage (user_id, format, (void *) &message)

Parameters

Name	Code	Type		Description
user_id	P1	int		MRCI assigned user id returned from MRCI_RegisterUser
format	P4	MRCI_MessageFormatType	*	Pointer to location for MRCI to return message format (e.g., TACFIRE, USMTF)
message	P4	void **		Pointer to location for MRCI to store message
msg_size	P4	int	*	Pointer to location for MRCI to store message size

Translator Common Modules
Modular Reconfigurable Message Translation (MRMT)

- MRMT takes a lifecycle approach with a flexible design to accommodate the addition of new message formats and the revision of currently utilized message formats.
- MRMT has Three Phases for preparation and use
  - Protocol Preparation

    Protocol Table generation routines and Parser modules are customized for a new message format.
  - Exercise Preparation

    Mapping Tables specifying the mapping of C4I to Simulation Messages are created by System Analysts for the particular messages utilized in the exercise. Translator Definition Language (TDL) files are prepared from the Mapping Table.
  - Initialization

    Message Structures are read from the Protocol Tables and Translation Objects are created from TDL files.

Generation of MRMT Protocol Tables

- To utilize a new message protocol, another module must be generated to take database or ASCII files describing the message structures and translate them into MRMT Protocol Tables

- We have currently done this for two protocols:

  – USMTF

  The following Ingress database files were used as the input to the USMTF Table Generation Module (current USMTF files are available on CD): fudbasic.ing, fudcolht.ing, fudname.ing, msgid.ing, msgmane.ing, msgsetor.ing, setfield.ing, settitle.ing, snrmks.ing, and setrmks.ing

  – CCSIL

  The following ascii files were used as the input to the CCSIL Table Generation Module : cfor.x and cfor_enum.x

MRMT Uniform Message Structure

- Protocol tables have data structures for messages, fields and enumerations
- An abstract description of the structures is:

Message_rec:	Field_rec:		Enumeration_rec:
name, serial_id, entry_node, next_node	label, control_flag, sequence, byte offset, min_size max_size	data_type msg_data msg_link_ptr leaf_ptr next_ptr repeat_on	name, list_size data_list next_node

MRMT Parsers

- Each Protocol must have a message Parser that can both put a message into the MRMT uniform message structures and construct a message from the uniform message structures.

- Currently, parsers for USMTF (character oriented) and CCSIL (binary) have been constructed.

- The character oriented parser and binary parser are customizable for similar protocols.

MRMT Mapping Tables

- Mapping Tables are the basic documentation on how to perform the mapping. They are the source documents for the TDL file.
- A portion of a Mapping File is given below:

CCSIL Message ID	CCSIL Field Name	CCSIL Data Type (1)	Source Protocol	Source Message ID	Source Set ID	Source Field ID	Category (2)	Data Type (3)	Frequency	Default Value	To (4)	From (4)
*6-Air Base Status Report**			*USMTF*	*ABSTAT*					*Triggered*		*AOC (CTAPS)*	*WOC (CTAPS W/S)*
					MSGID		*M*					
						(1) Title: ABSTAT	M	6A		ABSTAT		
	Unit ID*					(2) Originator: u name	M	1-20X		_		
					BASESTAT		*M*					
						(1) DE: Data Entry	M	2N		_		
	Base Name*					(2) Base Name	C	1-20X				
	Operational Status Code*	E				(4) OPS: Operational Status (entry list 70)	M	3A		entry list 70; conversion required to CTAPS required value R/Y/G (Red/Yellow/Green)		
	Time Of Report*					(5A) EFFTIME: Effective Day-Time (Day/Hour/Minute Time Zone)	M	7AN				
	Time to Return to Operational Status					(6A)TIME: Expected Time to Return to Operational Status (Day/Hour/Minute Time Zone)	O	7AN				

(1) To Be Determined by CCSIL Development Team
(2) Category Code: M=Mandatory, C=Conditional, O=Operational Context
(3) USMTF Data Type: A=Alpha[A-Z], N=Number, X=Alpha Numeric, S=Special Characters [? , . : ; ( ) / -]
(4) To & From: AB=Airbase, INTEL=Intelligence Services, WOC=Wing AOC=Air Operations Center

MRMT Translation

- The MRMT Translator Engine is Object Oriented and utilizes the Uniform Message Structures populated by the Parser.
- At initialization, the MRMT reads in the TDL file to create Translation Objects. Each Translation Object specifies how to translate message data to a target message field.
- The TDL file utilizes OP Codes to specify application of generic translation methods (such as mapping a string data type to a number data type, or coordinate transformations).
- At runtime, a method is invoked from the appropriate Translation Object to map to a field in the target message.

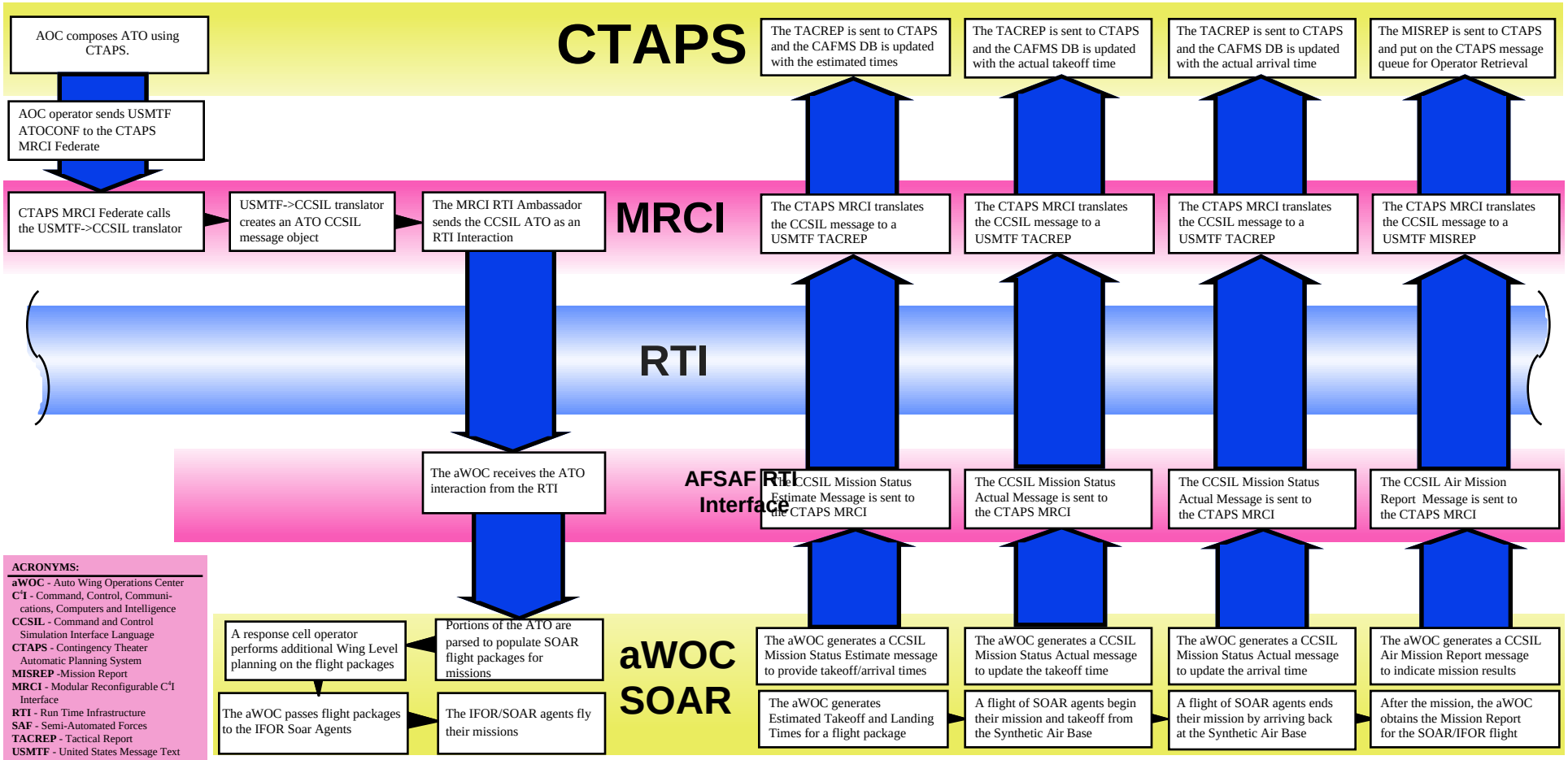
Sample MRMT TDL File

The following is a portion of a TDL file to map a CCSIL Unit Situation Report to a USMTF SITREP:

```
t|unit_situation_report_msg|SITREP
|3|2,EXER:2,def1|MRCI-CFOR|
|3|2,EXER:2,def2|SAIC-TRR, 12/15/97|
|3|2,MSGID:2,def1|SITREP|
|2|2,MSGID:2,def2|2,unit_data:2,unit_name|
|6|2,HEADING:2,def1|ENEMY|2,unit_data:2,number_of_enemy_data|>|0|
|7|2,5EUNIT:2,DE|1|
|2|2,5EUNIT:2,CY|2,unit_data:2,enemy_data:2,side|
|2|2,5EUNIT:2,ACTTYP|2,unit_data:2,enemy_data:2,activity:2,movement_status|
|4|2,5EUNIT:2,TIMPOS|2,unit_data:2,enemy_data:2,date_time:2,day|
   2,unit_data:2,enemy_data:2,date_time:2,hour|
   2,unit_data:2,enemy_data:2,date_time:2,minute|
   2,unit_data:2,enemy_data:2,date_time:2,time_zone|
|4|2,5EUNIT:2,UNITLOC|2,unit_data:2,enemy_data:2,location:2,latitude|
   2,unit_data:2,enemy_data:2,location:2,longitude|
|2|2,5EUNIT:2,ENUNIT|2,unit_data:2,enemy_data:2,echelon|
```

Prototype MRCI

C4I to SAF Message Flow



ACRONYMS:

- **aWOC** - Auto Wing Operations Center
- **C4I** - Command, Control, Communications, Computers and Intelligence
- **CCSIL** - Command and Control Simulation Interface Language
- **CTAPS** - Contingency Theater Automatic Planning System
- **MISREP** -Mission Report
- **MRCI** - Modular Reconfigurable C4I Interface
- **RTI** - Run Time Infrastructure
- **SAF** - Semi-Automated Forces
- **TACREP** - Tactical Report
- **USMTF** - United States Message Text Format

CESS Defined

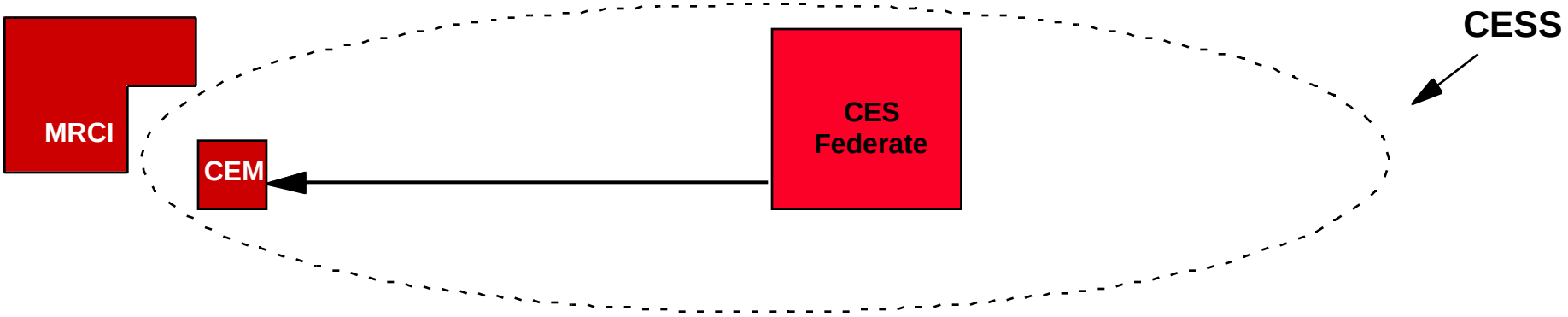
(pronounced CEASE)

The Communications Effects Server System (CESS) is a modular server system used to apply tactical communications effects in a simulated environment. The CESS operates within High Level Architecture (HLA) guidelines as defined by the Defense Modeling and Simulation Organization (DMSO).

The CESS is composed of two parts:

1) The Communications Effects Module (CEM)- A module incorporated within a system acting as a federate in an HLA exercise
2) The Communications Effects Server (CES)- A stand alone system acting as a federate in an HLA exercise

- Both components will communicate via the HLA Run Time Infrastructure (RTI).
- The Modular Reconfigurable C4I Interface (MRCI) will be the first system to integrate the CESS.



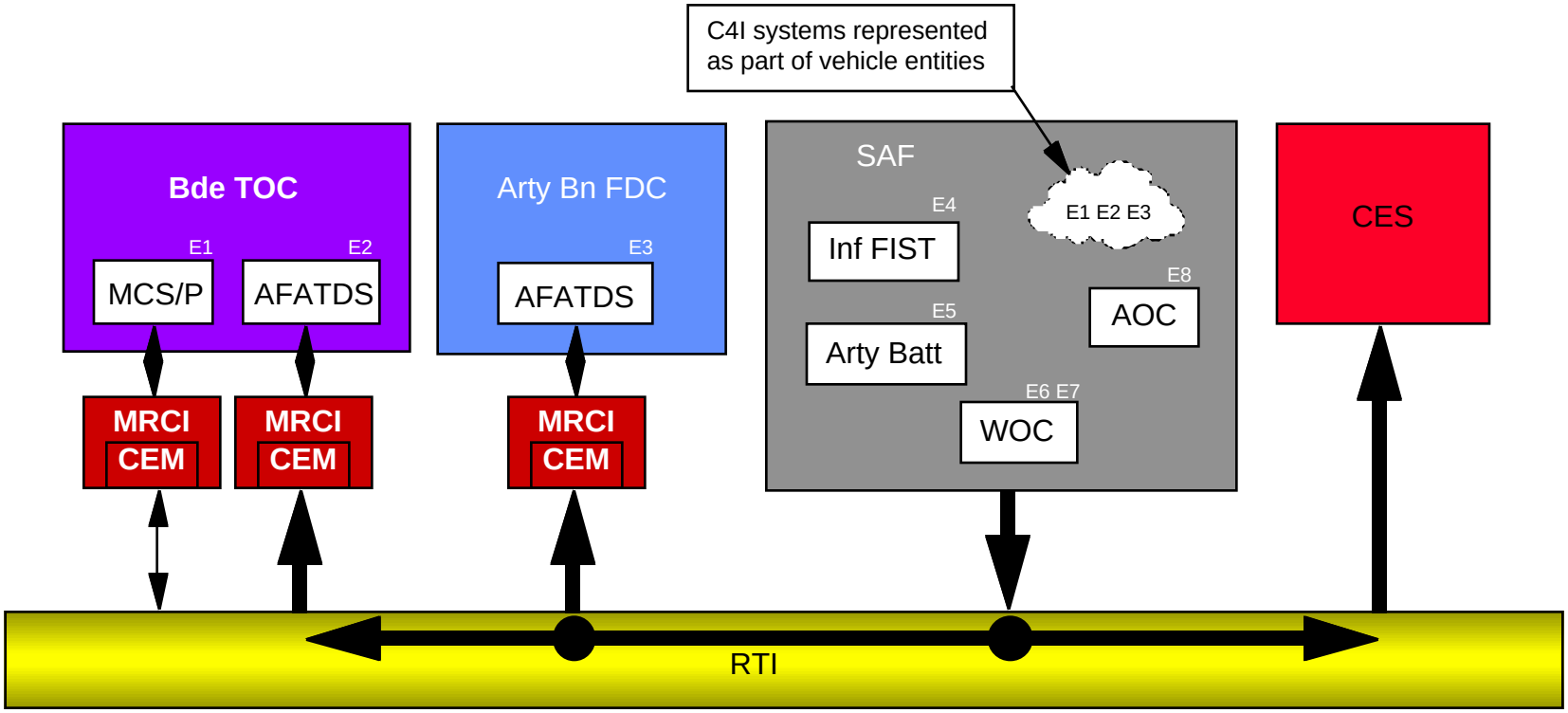
The CEM determines message delivery based on:
- communications object settings
  OR
- degradation parameters received from the CES

The CES:
- monitors aspects related to communications
  AND
- generates degradation parameters on a per message basis

Basic Concept- Part I

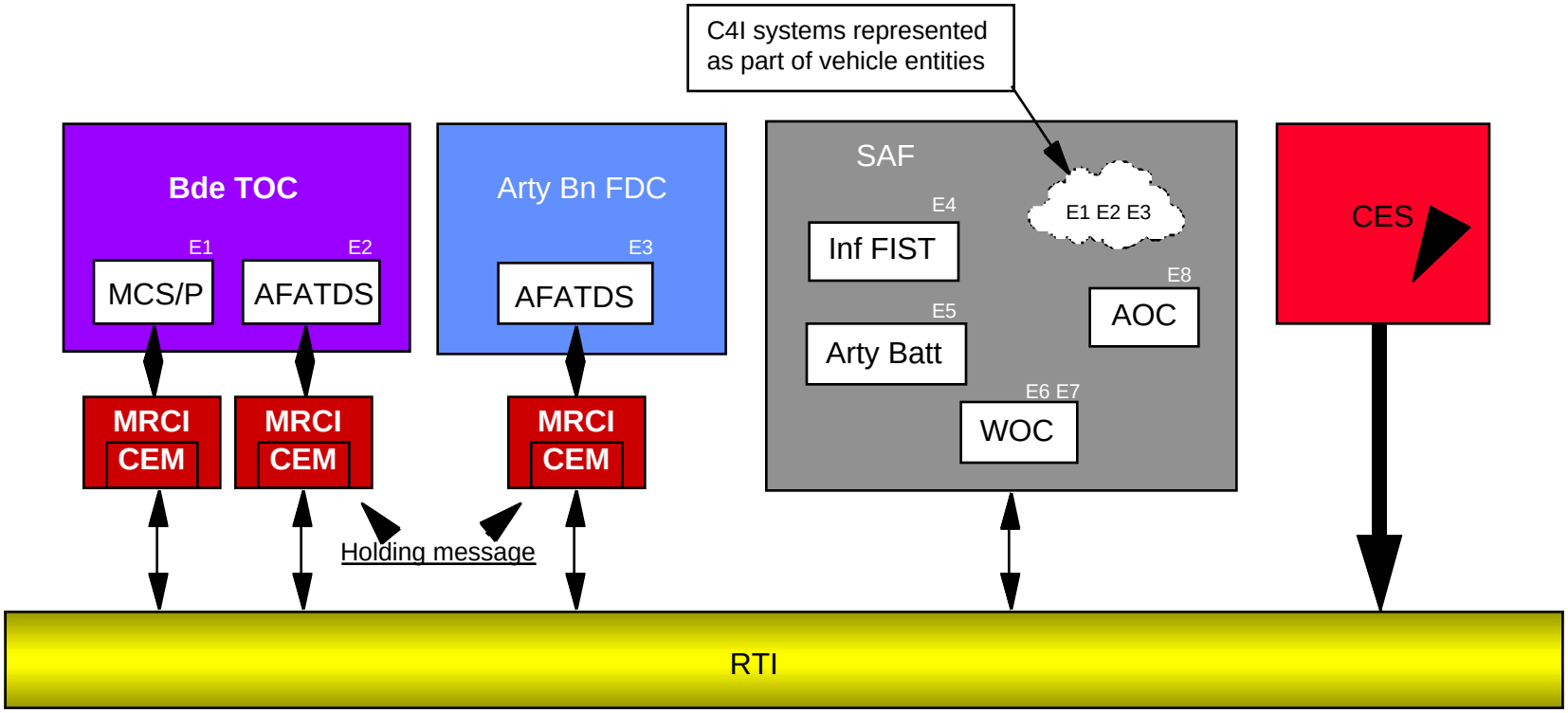
1) A message interaction is sent via the RTI
2) The message is received by federates subscribing to message interactions. Each federate determines if the message was intended for it. In this example, the message was intended for and received by:
   - The Bde TOC AFATDS system's representative MRCI
   - The Arty Bn FDC AFATDS system's representative MRCI

   But, because the Communications Effects Server (CES) is interested in all message interactions, it receives the interaction as well.



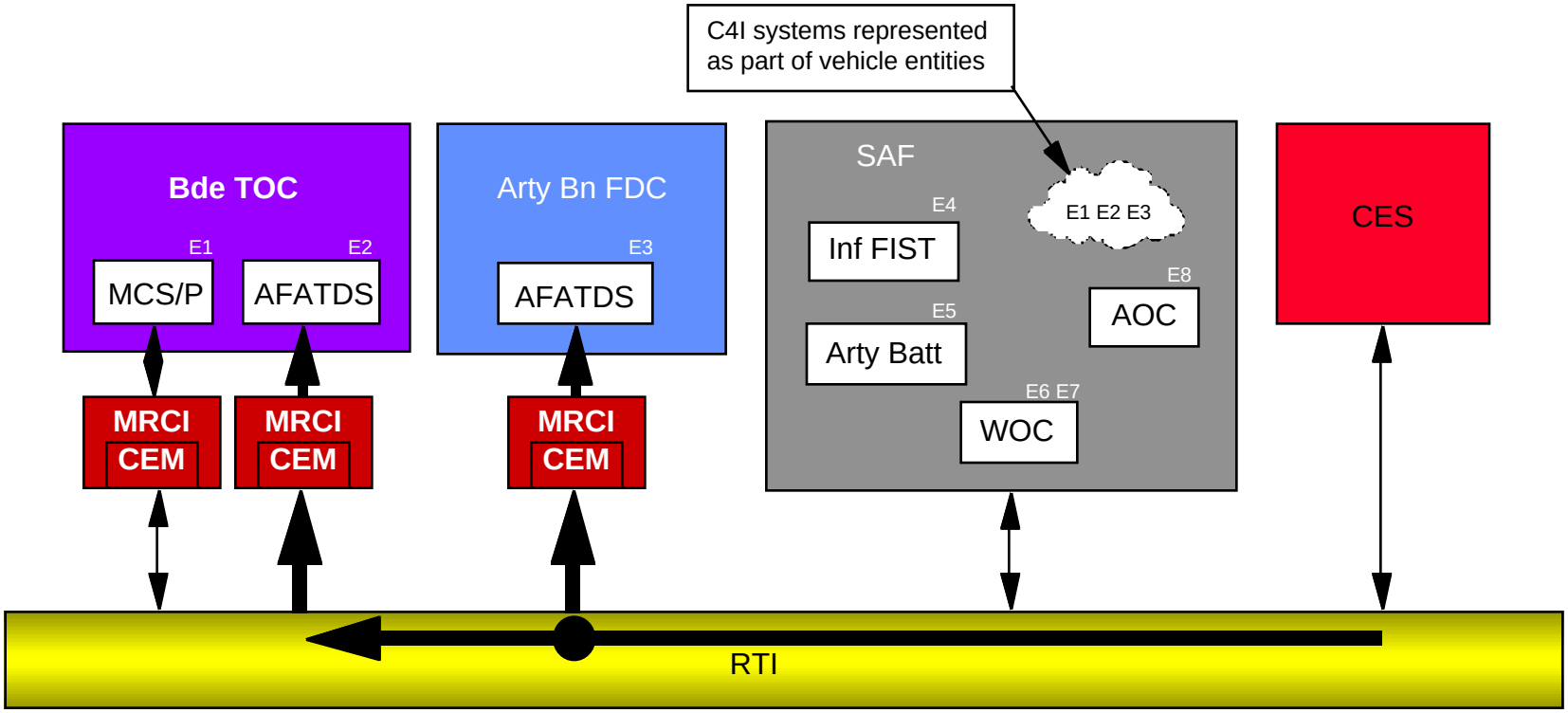
Basic Concept- Part II

3) Each Communications Effects Module (CEM- incorporated within each MRCI) holds the message until either:
   a) It receives a Latency Time (LT) from the Communications Effects Server (CES)
      OR
   b) A maximum LT value expires, in which case the message is released (in this example the max LT value does not expire).
4) The CES runs the message through its communications model and generates an expected time of arrival (referred to as the LT). It then sends the LT (along with unique message and receiver identifiers) as an interaction via the RTI.



Basic Concept- Part III

5) The LT interaction is received by every subscribing MRCI. It is only used when the held message corresponds to the LT's receiver and message ID.
6) The CEM (resident in the MRCI) releases the message when the LT expires.



The Three Key Representational Levels Within the CESS

Static without degradation

Communications Object

Level 1

Purpose:
- Adequately represents communications device
- Facilitates dissemination of changes in attributes

Used by:
- CES as communications determinant
- CEMs as most basic communications determinant

Maintained and updated by:
- Creating federate upon initialization and attribute changes

Dynamic periodic

Max. Latency Time (LT) Matrix Interaction

Level 2

Purpose:
- Sender/receiver pair locates time value in matrix- used as coarse time release value
- Backup for LT interaction (if processing + transport of LT > max. time value then release message

Used by:
- CEM as coarse message release determinant

Generated and sent by:
- CES upon initialization and when significant change in comm state occurs

Dynamic per message

Latency Time (LT) Interaction

Level 3

Purpose:
- Contains value that accurately represents simulated time of message delivery
- Determines when CEM releases message

Used by:
- CEM to accurately determine message release time

Generated and sent by:
- CES upon receipt of message

Examples of The 3 Levels

Communications Object

Object	Attribute	Datatype
Radio Comm	Identification	Transmitter-ID-Type
	Device_State	Transmit_State_Enum
	Absolute_Antenna_Locatio	Point-Location
	Relative_Antenna_Locatio	Point-Location
	Antenna_Pattern	Antenna-Pattern-Type
	Power	Float
	Frequency	Float
	Transmit_Freq_Bandwidth	Float
	Modulation	Modulation-Type
	Crypto System	Crypto_System_Enu
	Crypto Key ID	Crypto-Key-ID-Type
	Apply_Degradation	Boolean
	Hopset	Integer
	Bandwidth	Integer
	Packet_MTU	Integer
	Net_Access_Time	Integer
	Block_Mode	Integer
Wire Comm	Identification	Transmitter-ID-Type
	Device_State	Transmit_State_Enum
	Absolute_Location	Point-Location
	Relative_Location	Point-Location
	Network_Access_Point_K	Transmitter-ID-Type
	Wire_Layout	Wire_Layout_Enum
	Wire_Info	Wire-Info-Type
	Retransmit_Info	Retransmit-Type
	Framing_Method	Frame_Enum
	Network_Protocol	Network_Enum
	Crypto System	Crypto_System_Enu
	Crypto Key ID	Crypto-Key-ID-Type
	Apply_Degradation	Boolean
	Bandwidth	Integer
	Packet_MTU	Integer
Comm Device Obj	Identification	Transmitter-ID-Type
	Device_State	Transmit_State_Enum
	Absolute_Location	Point-Location
	Relative_Location	Point-Location
	Device_Class	Device_Class_Enum
	Network_Connection_Tabl	Alphanumeric

Max. Latency Time (LT) Matrix Interaction

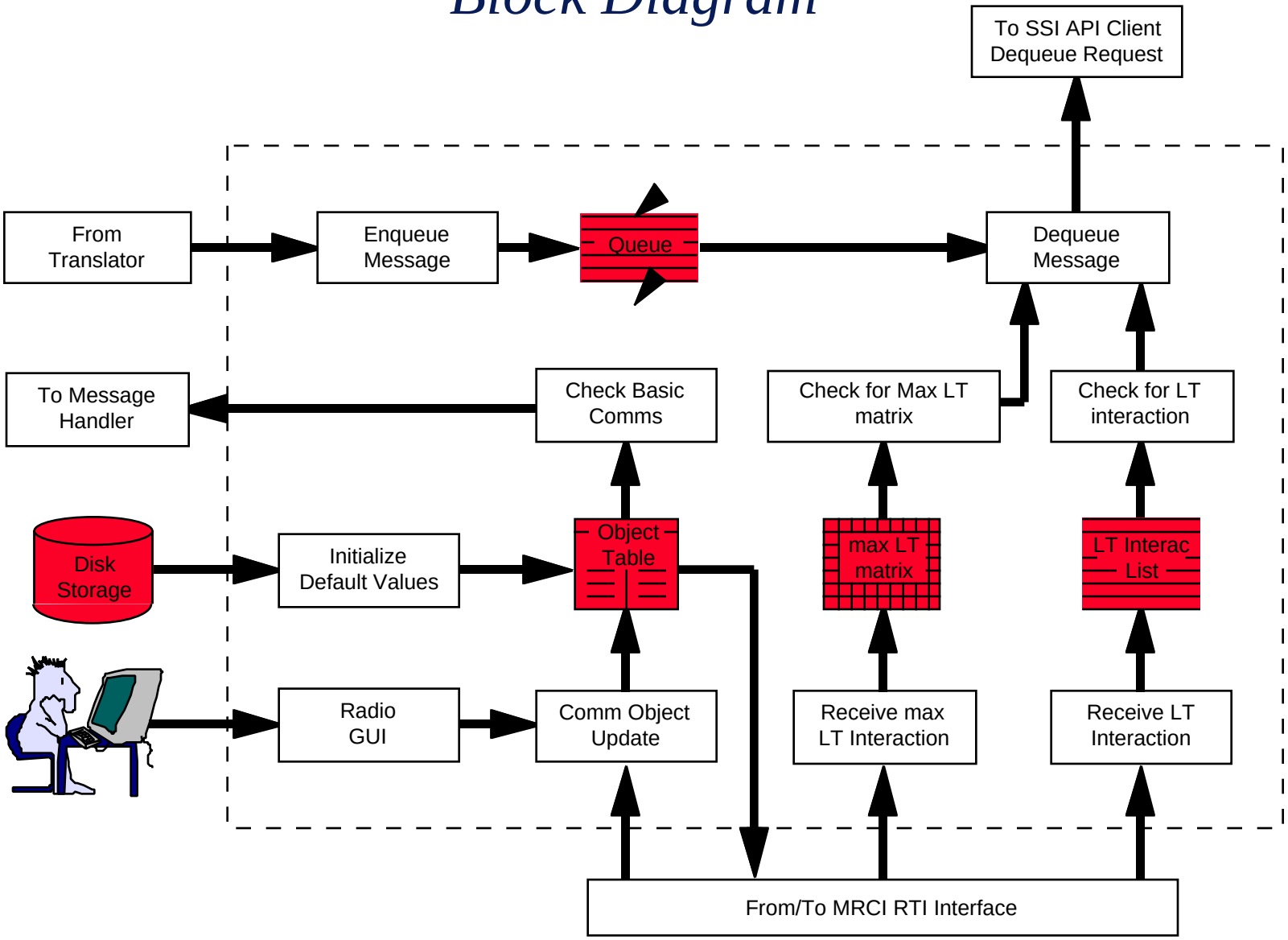
Max Latency Time (Sec)	Receiver 1	Receiver 2	Receiver 3	Receiver 4	Receiver ...
Sender 1		2	0	0	x
Sender 2	2		3	2	x
Sender 3	0	1		2	x
Sender 4	0	2	2		x
Sender ...	x	x	x	x	

Latency Time (LT) Interaction

LATENCY TIME	Message_ID
	Receiver_ID
	Latency_Time

CEM Data Model

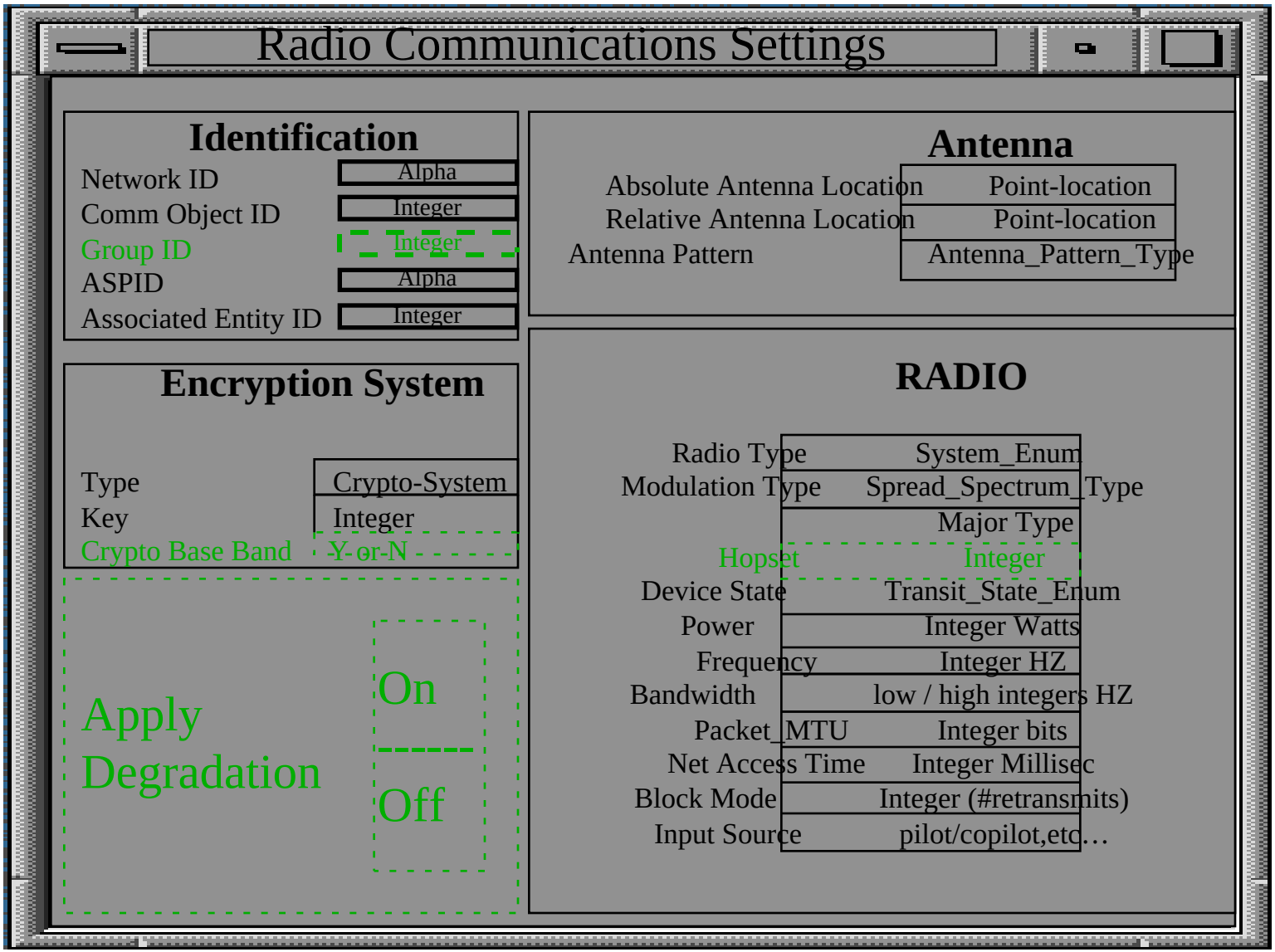
Block Diagram



CEM GUI

Used by SAF

CESS Addition



System Dictionary

Term	Abbreviation	Definition
Communications Effects Server System	CESS	The general name given to the modular server system used to apply tactical communications effects in a simulated environment.
Communications Effects Server	CES	A system that monitors aspects related to communications and generates degradation parameters on a per message basis.
Communications Effects Module	CEM	A module incorporated within a system that determines message delivery based on communications object settings or degradation parameters received from the CES.
Latency Time	LT	The delayed time of delivery of a message (base time = RTI federation time). Sent by CES as an interaction and received by CEM.
Maximum Latency Time Matrix	Max LT matrix	A matrix populated by values representing the maximum latency incurred on message delivery. Values are identified by sender/receiver pairs.
Maximum Latency Time Value	Max LT value	The value obtained when a sender/receiver pair is applied to the max LT matrix.

MRCI Support of Multiple RTI Versions

- **Modular generic interface currently supports the STOW RTI A version**
- **Prepared to upgrade to Beta version by 31st January**
- **Will include F.0 by early February**
- **Both STOW and Familiarization RTI versions will be selectable MRCI options**

MRCI RTII Reusability: I/ITSEC Demo

- **Scenario**
  - MCS/P tasking a CCTT SAF unit
  - CCTT SAF unit reporting to MCS/P
- **RTI Interface Evolution**
  - Original prototype used RTI version 0.33
  - Transitioned to STOW RTI A.1 for I/ITSEC
- **Reused MRCI RTI Interface and parts of Translation modules on CCTT SAF simulation side of RTI**

HLA Functionality Employed

- **FOM**
  - CCSIL Interactions
  - Communications Objects
- **Services**
  - Federation Mgmt (Create & Join Federation)
  - Declaration Mgmt (Publish & Subscribe Objects and Interactions)
  - Object Mgmt (Request_ID, Register Object,
    Send Interaction, Receive Interaction)

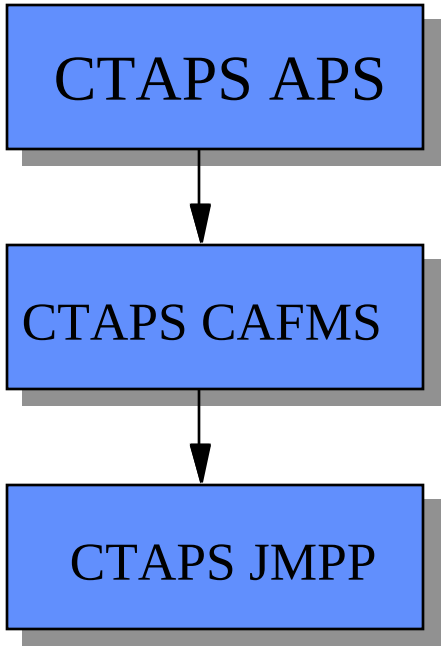
TRR Agenda (1 of 3)

Time	Subject	
0830-0840	Introductions and Program Status	- Park
0840-0930	MRCI Design Update	
	- MRCI Application Programmers Interface	- Griggs
	- MRCI Common Modules	- Hieb/ Silva
	- MRCI Run Time Infrastructure Interface	-McKenzie
0930-0950	CTAPS Update	- Bretton
	- SSI Implementation	
	- Mission Threads / Messages	- Ashley
0950-1000	Break	

CTAPS/MRCI SSI
(System Specific Interface)

- The CTAPS/MRCI SSI will allow CTAPS to send/receive USMTF messages to/from the simulation environment via MRCI.
- The CTAPS SSI does not interfere with CTAPS operation.
- The CTAPS SSI interfaces to the CTAPS using already existing CTAPS communication features.

CTAPS Operation



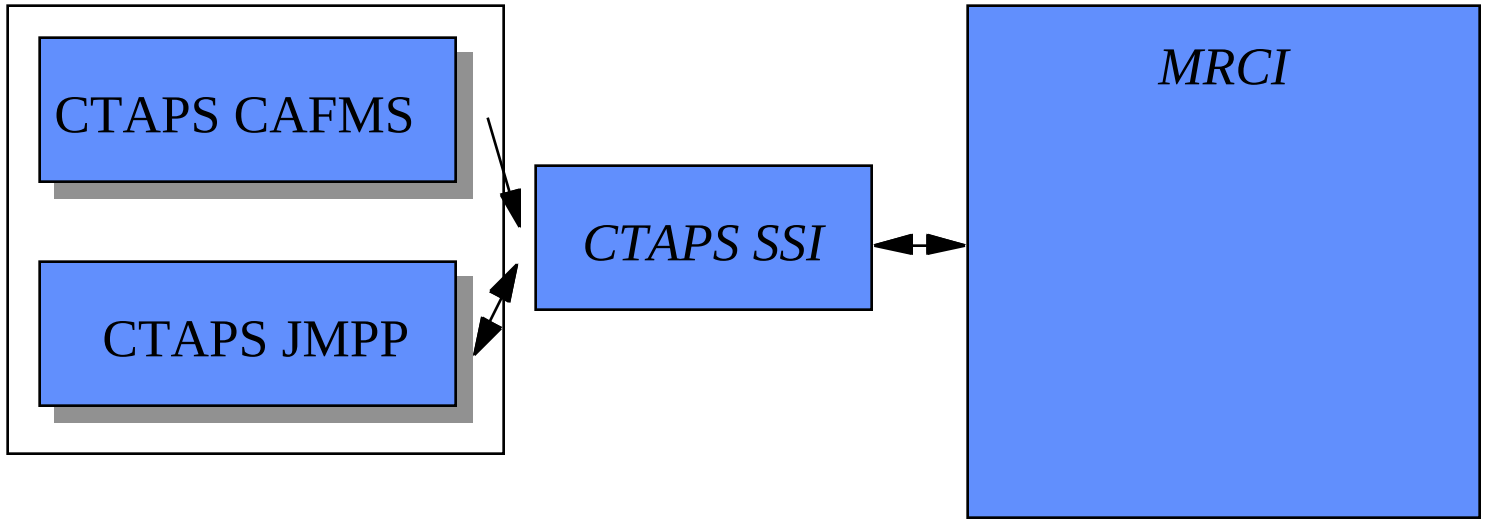
* Advanced Planning System (APS) “plans” the ATO and forwards ATO data to CAFMS

* CAFMS “collates” the ATO and creates an ATOCONF USMTF (text) message. Sends ATOCONF to JMPP for release via CTAPS communication. ATOCONF can be sent to other CTAPS sites at this time.

* JMPP adds information required to send ATOCONF via AUTODIN.

* Note that a copy of the ATO is kept with CAFMS and another (JANAP 128) copy is kept with JMPP.

The CTAPS SSI



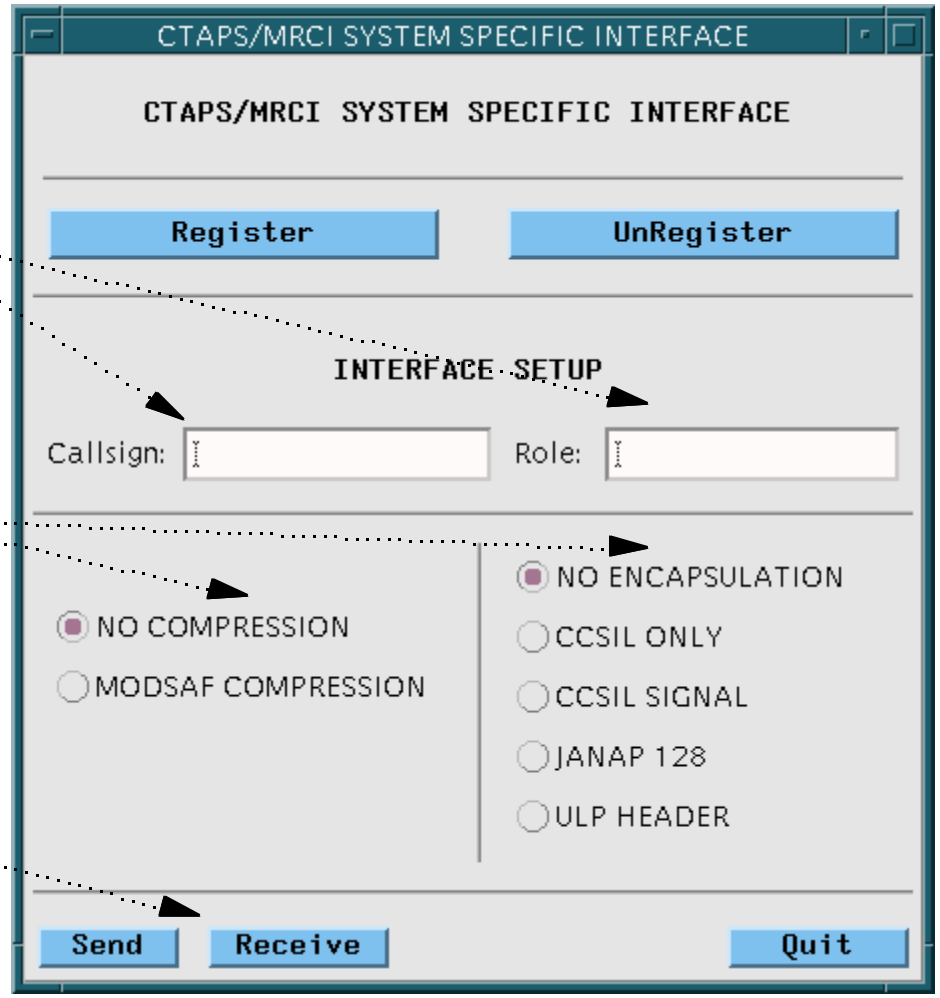
* The CTAPS SSI can get an ATO from either system.
* The SSI can get *and* receive messages with JMPP.

The CTAPS SSI GUI (1 of 3)

* The user specifies *Callsign* and *Role* before Registering with the **MRCI Server**.

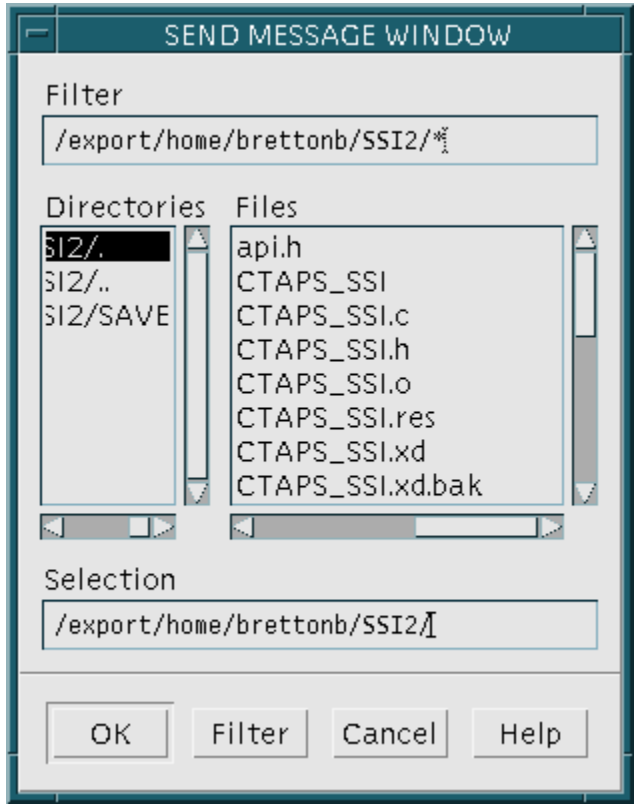
* The user can also specify message *compression* and *encapsulation* types.

* The SSI can then Send and Receive messages.



The CTAPS SSI GUI (2 of 3)

* A standard file selection tool is used to select the message to send from CTAPS to MRCI.

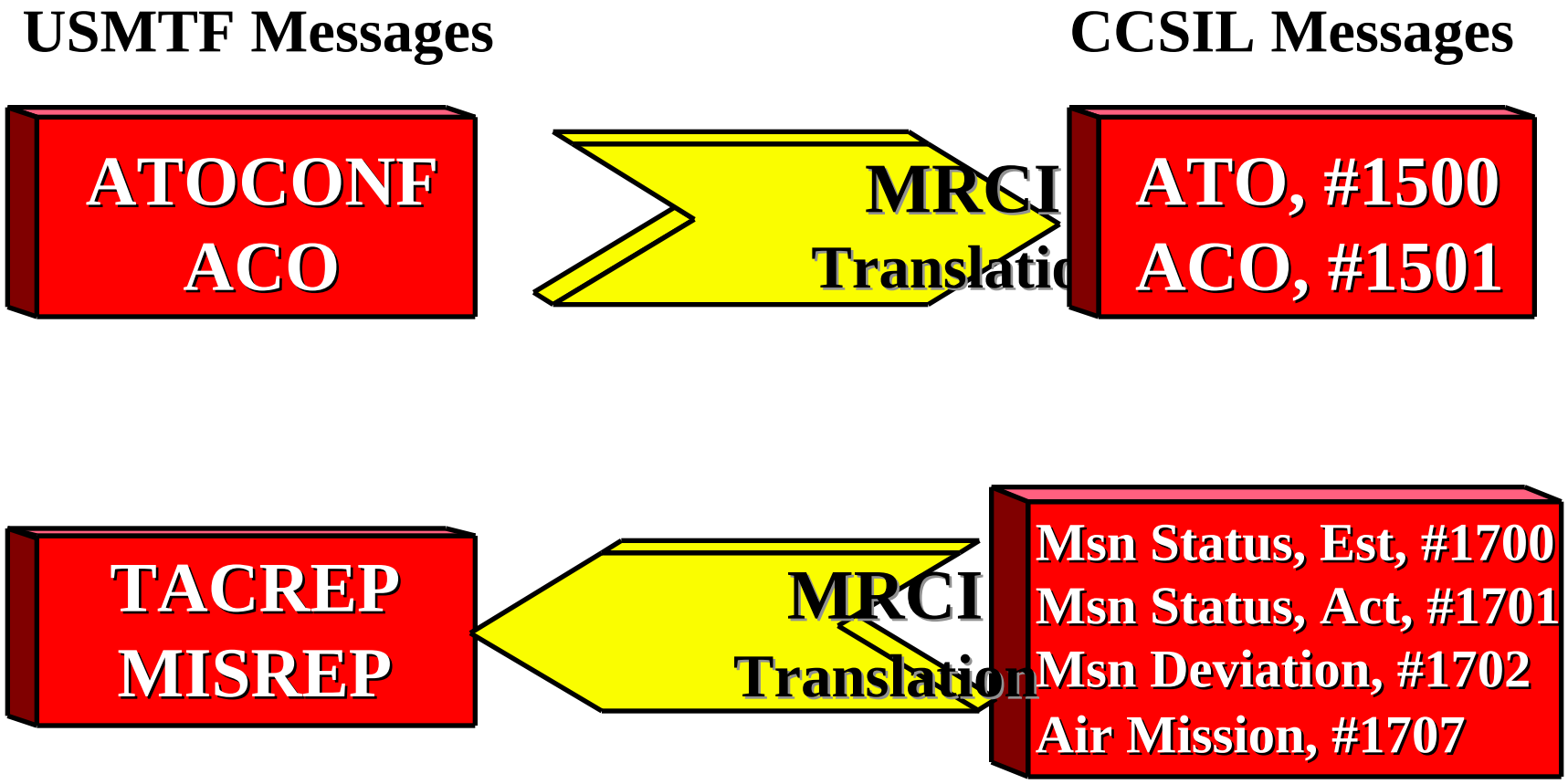


The CTAPS SSI GUI (3 of 3)

* When a message is received from MRCI, it is displayed in a window for the CTAPS SSI user to see.

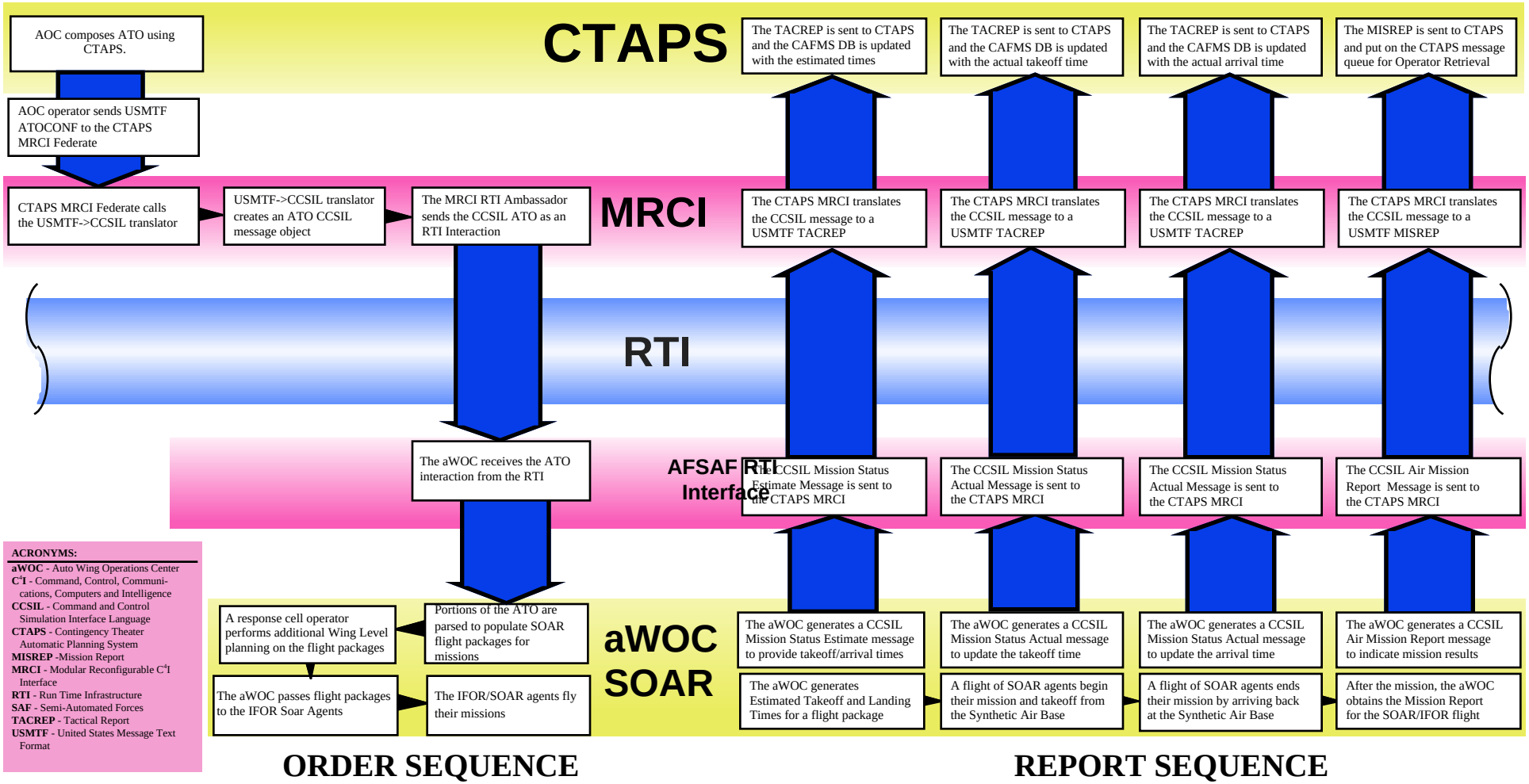
* The user will be able to save the message to any location.

MRCI CTAPS Messages



Prototype MRCI

C4I to SAF Message Flow



ACRONYMS:

- **aWOC** - Auto Wing Operations Center
- **C4I** - Command, Control, Communications, Computers and Intelligence
- **CCSIL** - Command and Control Simulation Interface Language
- **CTAPS** - Contingency Theater Automatic Planning System
- **MISREP** -Mission Report
- **MRCI** - Modular Reconfigurable C4I Interface
- **RTI** - Run Time Infrastructure
- **SAF** - Semi-Automated Forces
- **TACREP** - Tactical Report
- **USMTF** - United States Message Text Format

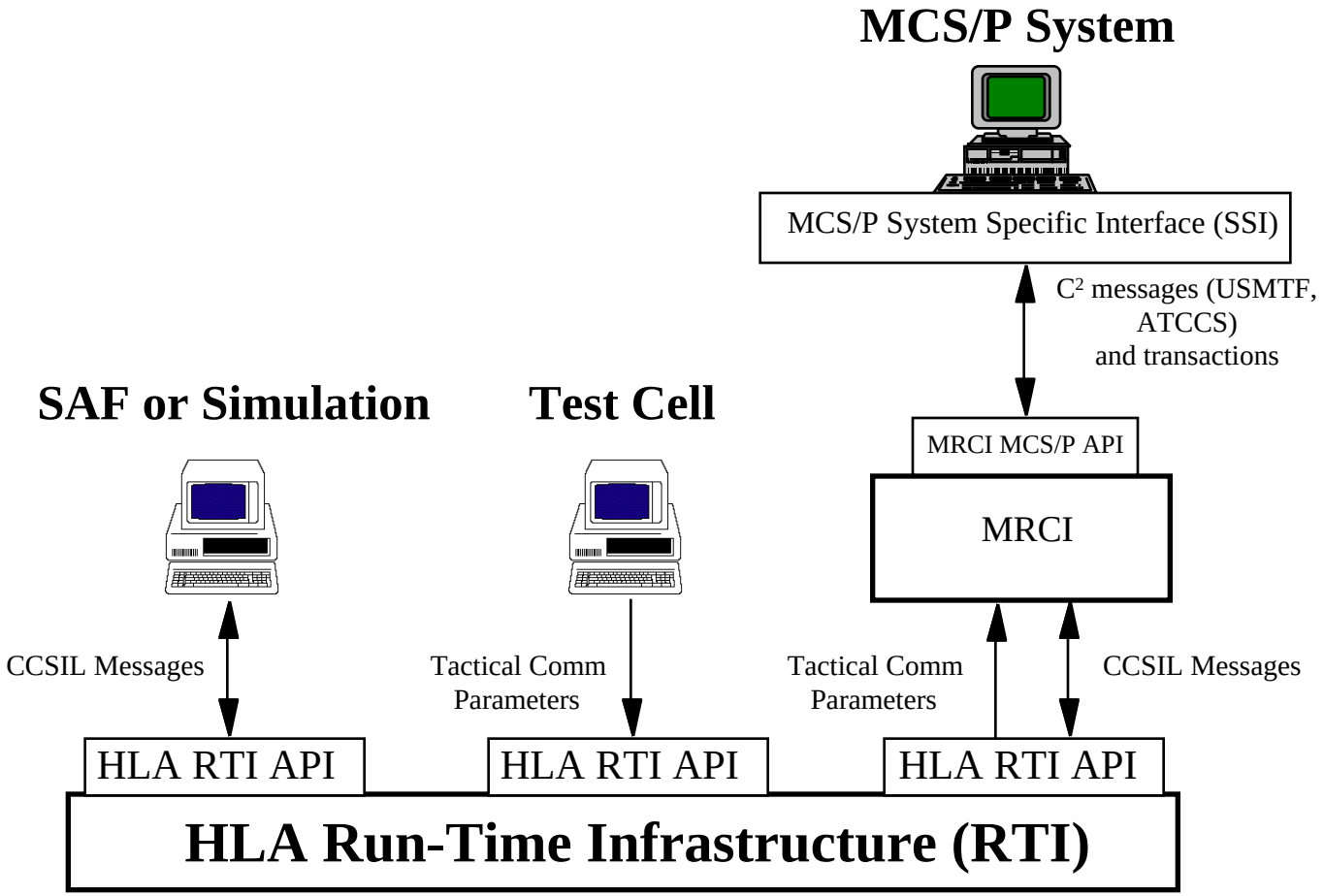
TRR Agenda (1 of 3)

Time	Subject	
0830-0840	Introductions and Program Status	- Park
0840-0930	MRCI Design Update	
	- MRCI Application Programmers Interface	- Griggs
	- MRCI Common Modules	- Hieb/ Silva
	- MRCI Run Time Infrastructure Interface	-McKenzie
0930-0950	CTAPS Update	- Bretton
	- SSI Implementation	
	- Mission Threads / Messages	- Ashley
0950-1000	Break	

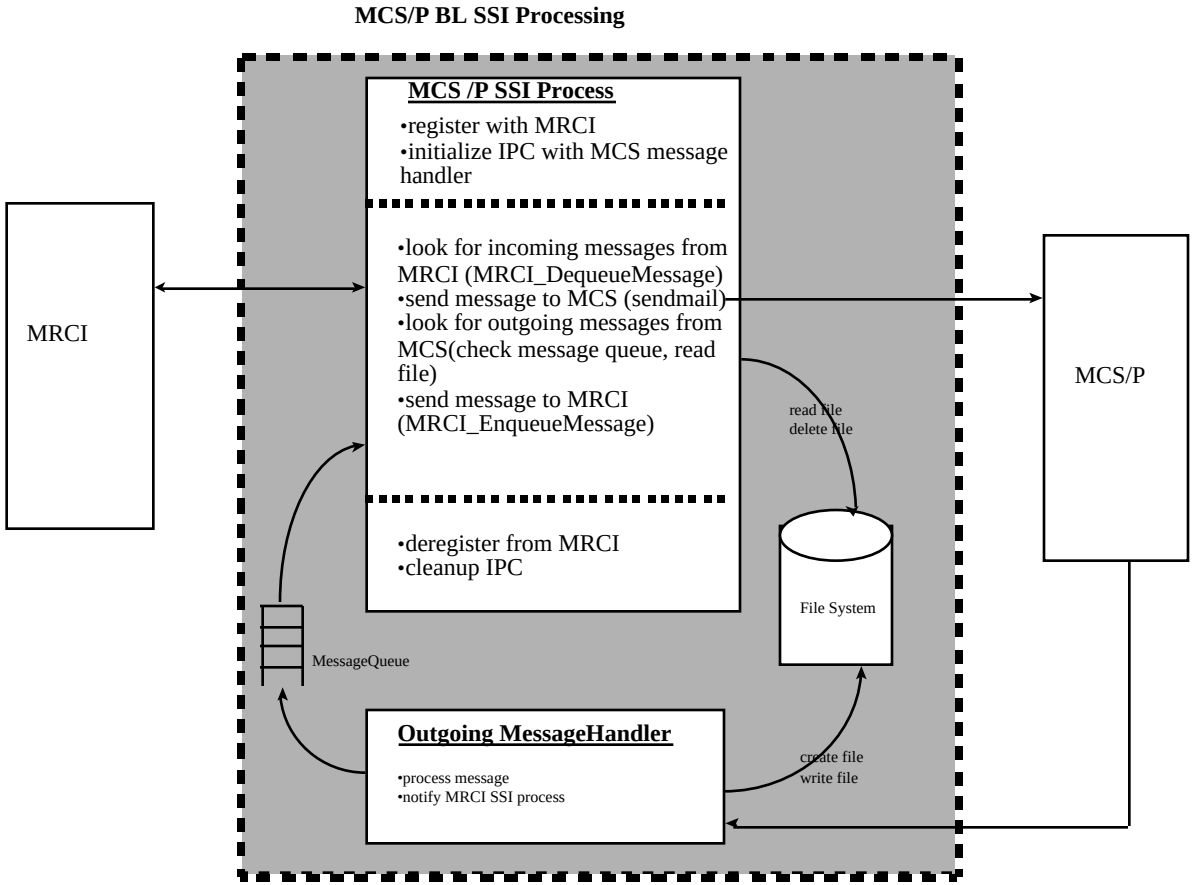
TRR Agenda (2 of 3)

Time	Subject	
1000-1020	MCS/P Update	- Howard
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1020-1040	AFATDS Update	- Anglin
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1040-1100	Simulation Federate Update	- Hieb
1100-1120	Test Program	- Chen
	- CT-5	
	- Post February MRCI Assessment Opportunities	
	- CBS JTC Update	

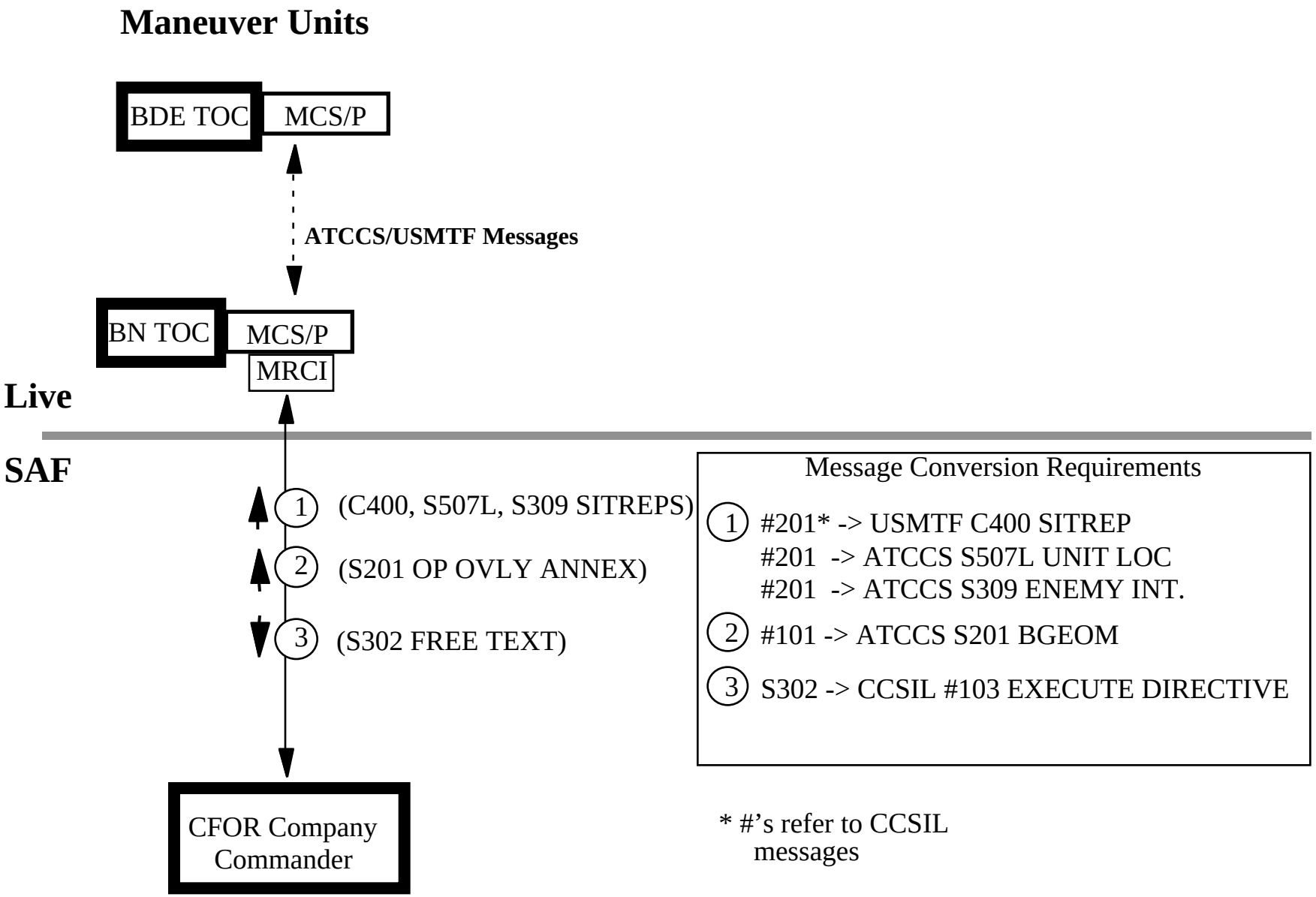
MCS/P SSI Architecture



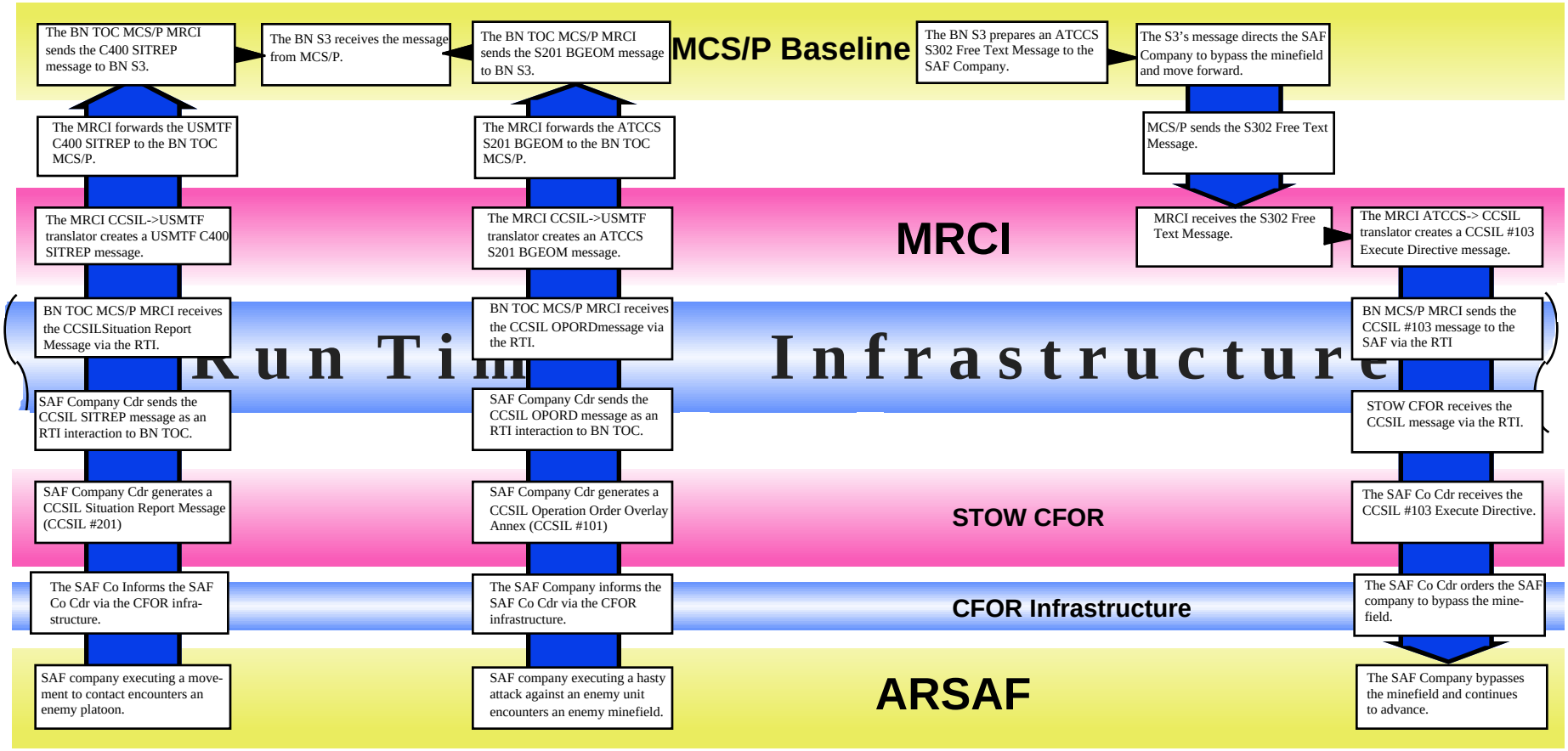
MCS/P SSI Design



ARSAF-MCS/P Message Interaction



MCS/P - ARSAF
Message Interaction



REPORT SEQUENCE OF ACTIVITY

ORDER SEQUENCE OF ACTIVITY

MCS/P Message Flow

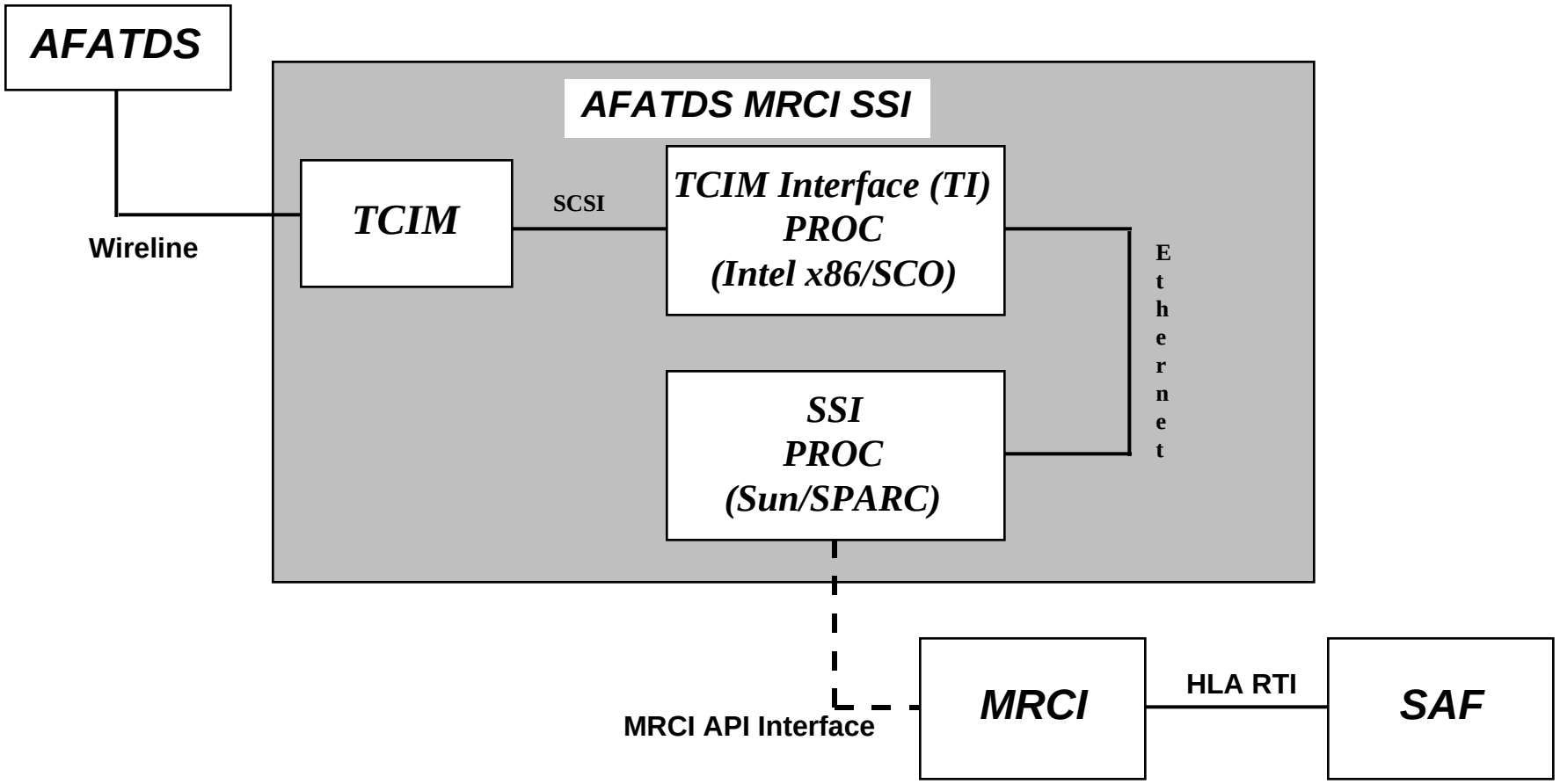
- **OUTBOUND**
  - => MCS/P Message Handler
  - => Outbound Message Queue
  - => Identify MRCI Messages
  - => Send Message to MRCI Enqueue Function
- **INBOUND**
  - => Check MRCI SSI Dequeue
  - => Read Message
  - => Forward Message to MCS (Sendmail)
  - => Process as Normal USMTF/ATCCS Message

TRR Agenda (2 of 3)

Time	Subject	
1000-1020	MCS/P Update	- Howard
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1020-1040	AFATDS Update	- Anglin
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1040-1100	Simulation Federate Update	- Hieb
1100-1120	Test Program	- Chen
	- CT-5	
	- Post February MRCI Assessment Opportunities	
	- CBS JTC Update	

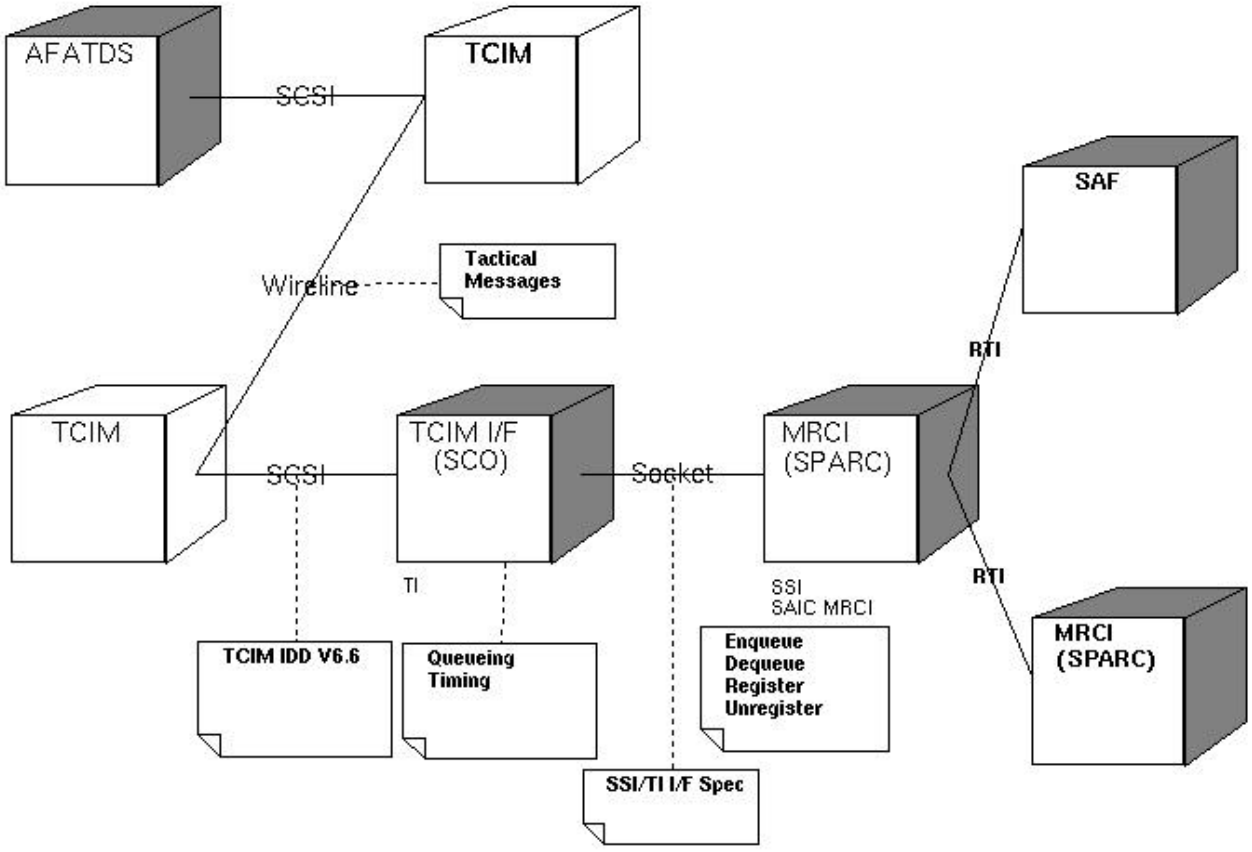
MRCI - AFATDS SSI

(functional location)



MRCI - AFATDS SSI

(physical connectivity among components)



MRCI - AFATDS SSI

(processor functions)

SSI PROC

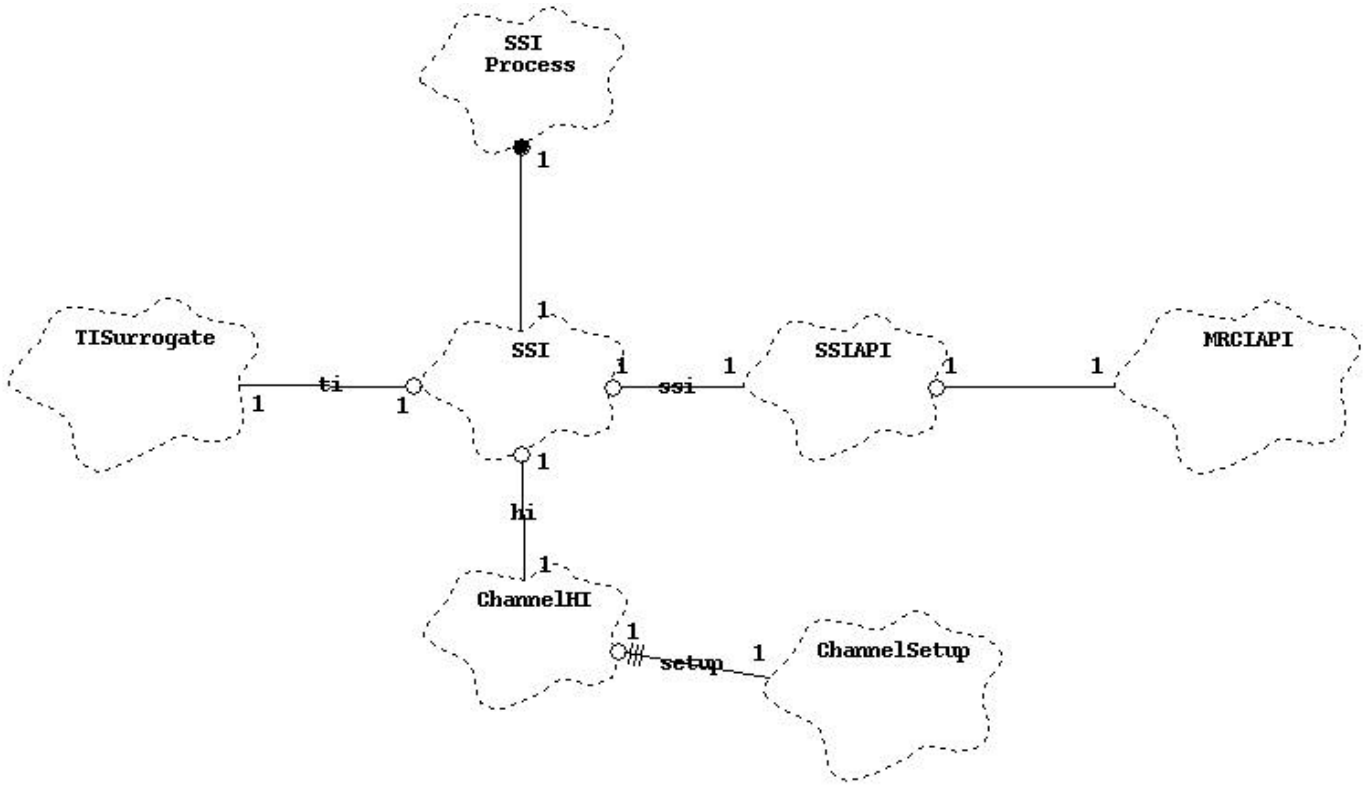
- Interface with MRCI API
- Configure via GUI
- Interface with TI

TI PROC

- Interface with TCIM via SCSI
- Protocol SW Management
- Protocol formatting
- Interface with SSI

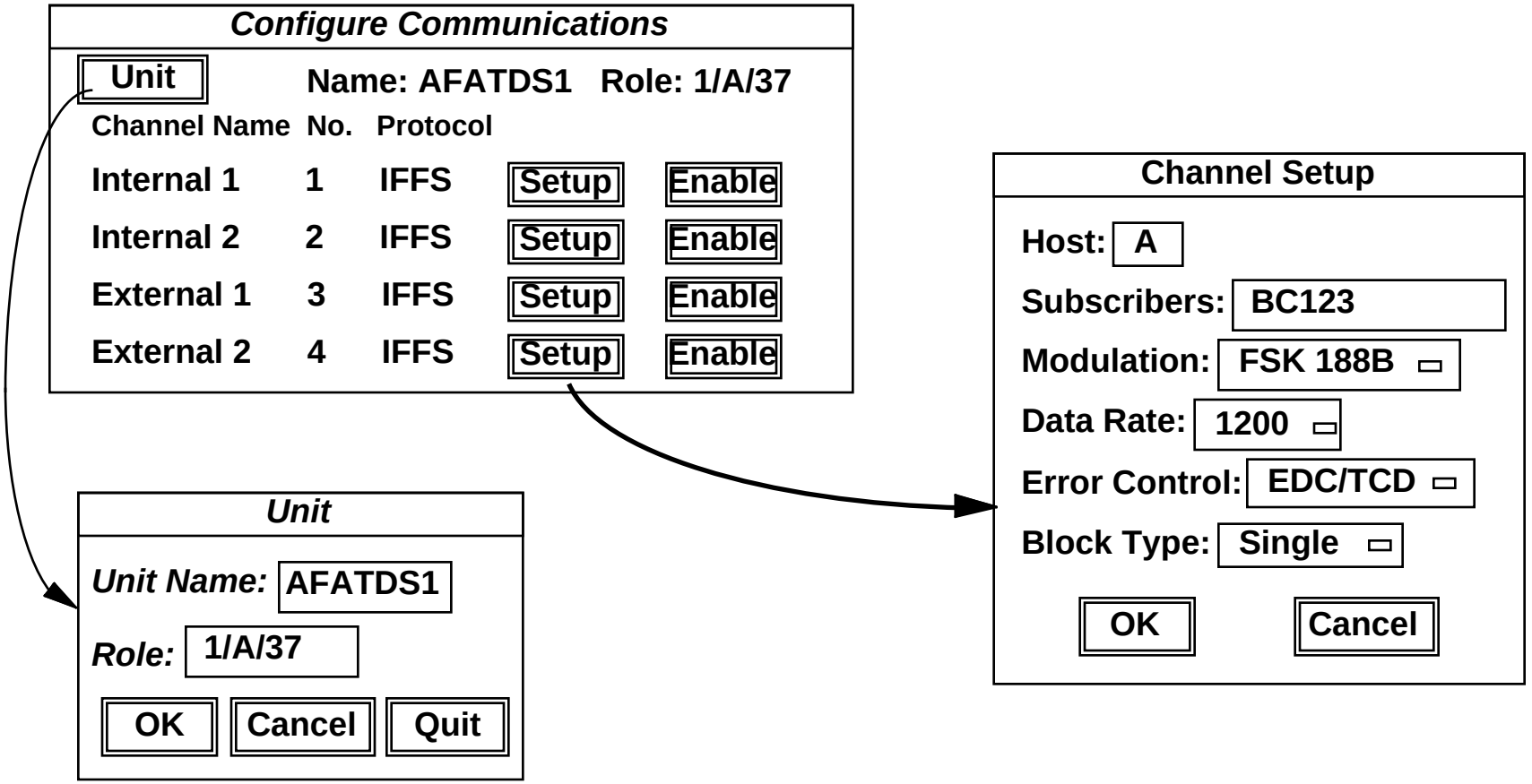
MRCI - AFATDS SSI

(SSI Design)

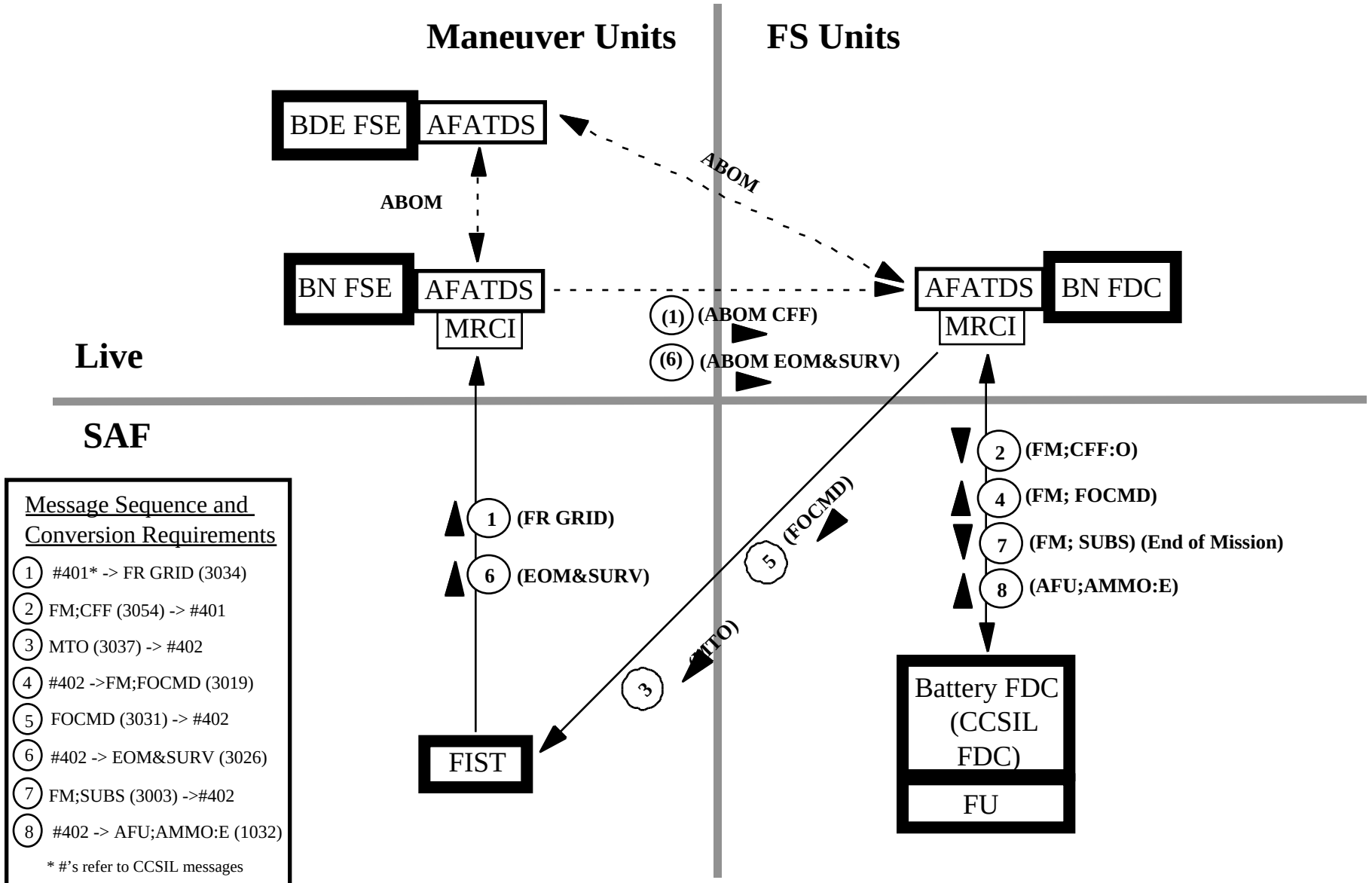


MRCI - AFATDS SSI

(GUI)



ARSAF-AFATDS Message Interaction



Message Sequence and Conversion Requirements

1. #401* -> FR GRID (3034)
2. FM;CFF (3054) -> #401
3. MTO (3037) -> #402
4. #402 ->FM;FOCMD (3019)
5. FOCMD (3031) -> #402
6. #402 -> EOM&SURV (3026)
7. FM;SUBS (3003) ->#402
8. #402 -> AFU;AMMO:E (1032)

* #'s refer to CCSIL messages

Please refer to separate large single page handout entitled "AFATDS - ARSAF Message Interactions"

TRR Agenda (2 of 3)

Time	Subject	
1000-1020	MCS/P Update	- Howard
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1020-1040	AFATDS Update	- Anglin
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1040-1100	**Simulation Federate Update**	**- Hieb**
1100-1120	Test Program	- Chen
	- CT-5	
	- Post February MRCI Assessment Opportunities	
	- CBS JTC Update	

Simulation Federate ARSAF

- Developer: SAIC-Burlington for Command Entity Software (Logicon for Army Knowledge Acquisition)
- Sponsoring Government Agency: DARPA
- Brief Description of Simulation: ARSAF is an integration of ModSAF, the CFOR infrastructure and Command Entity Reasoner Software to add explicit, virtual representation of command nodes, command & control information exchange and command decision-making to the simulation of Army individual platforms and small units
- For Use In: STOW97
- C4I Systems Interfaced To: MCS/P & AFATDS
- Status: The Automated Company Commander Command Entity has participated in STOW97 combined tests and is under continued enhancement. ARSAF is integrating the STOW RTI releases. Fire Support Entities are under development as is an Automated Battalion Commander Command Entity .

Simulation Federate
EAGLE

- Developer: MITRE
- Sponsoring Government Agency: TRAC
- Brief Description of Simulation: The Eagle system is an aggregate simulation at the Corps/Division level that simulates ground combat at the Company and Battalion level. Eagle is a combat analysis tool used for combat development studies. It is used in analyzing the effects of weapons systems, command and control, military doctrine, and organization on force effectiveness. The Eagle system is implemented in LISP on Sun workstations and runs faster than real-time.
- For Use In: DMSO/JSIMS HLA C2 Experiments
- C4I Systems Interfaced To: MCS/P & AFATDS
- Status: Eagle has been widely utilized within the Army Community over the past 5 years for analysis. It is currently fully integrated with RTI F.0 and has previously participated in the JTF Protofederation. It models Command Posts from Battalion to Corps.

Simulation Federate
AFSAF

- Developer: University of Michigan for SOAR & aWOC, Air Force Institute of Technology for the Airbase Model
- Sponsoring Government Agency: DARPA, ESC/AVM (PM)
- Brief Description of Simulation: AFSAF is an integration of SOAR/IFOR for simulation of pilots and ModSAF for providing an aircraft simulation capability. An Automated Wing Operations Center (aWOC) is used to manage and route communications among an existing AFIT airbase model, AFSAF and CTAPS
- For Use In: STOW97
- C4I System Interfaced To: CTAPS
- Status: AFSAF SOAR/IFOR has participated in STOW-E and STOW97 combined tests and is adding new pilot behaviors. The Base Model has been rehosted to a workstation environment (in FORTRAN). The aWOC is under development and scheduled for initial delivery for MRCI testing on January 15, 1997.

Simulation Federate
NASM/AP

- Developer: CACI (NASM follow-on in source selection)
- Sponsoring Government Agency: ESC/AVM (PM)
- Brief Description of Simulation: NASM/AP models air operations simulating aircraft missions, airbase conditions, logistics, sensor detection, ground-to-air and air-to-air engagements, theater ballistic missiles, and limited simulation of C4I elements including WOCs, ASOCs, & CRC/Es.
- For Use In: DMSO/JSIMS HLA C2 Experiments
- C4I System Interfaced To: CTAPS
- Status: NASM/AP is an advanced prototype of the NASM that will be part of the JSIMS program and participated in the JTF Protofederation. As a prototype, NASM/AP has limited functionality in some areas.

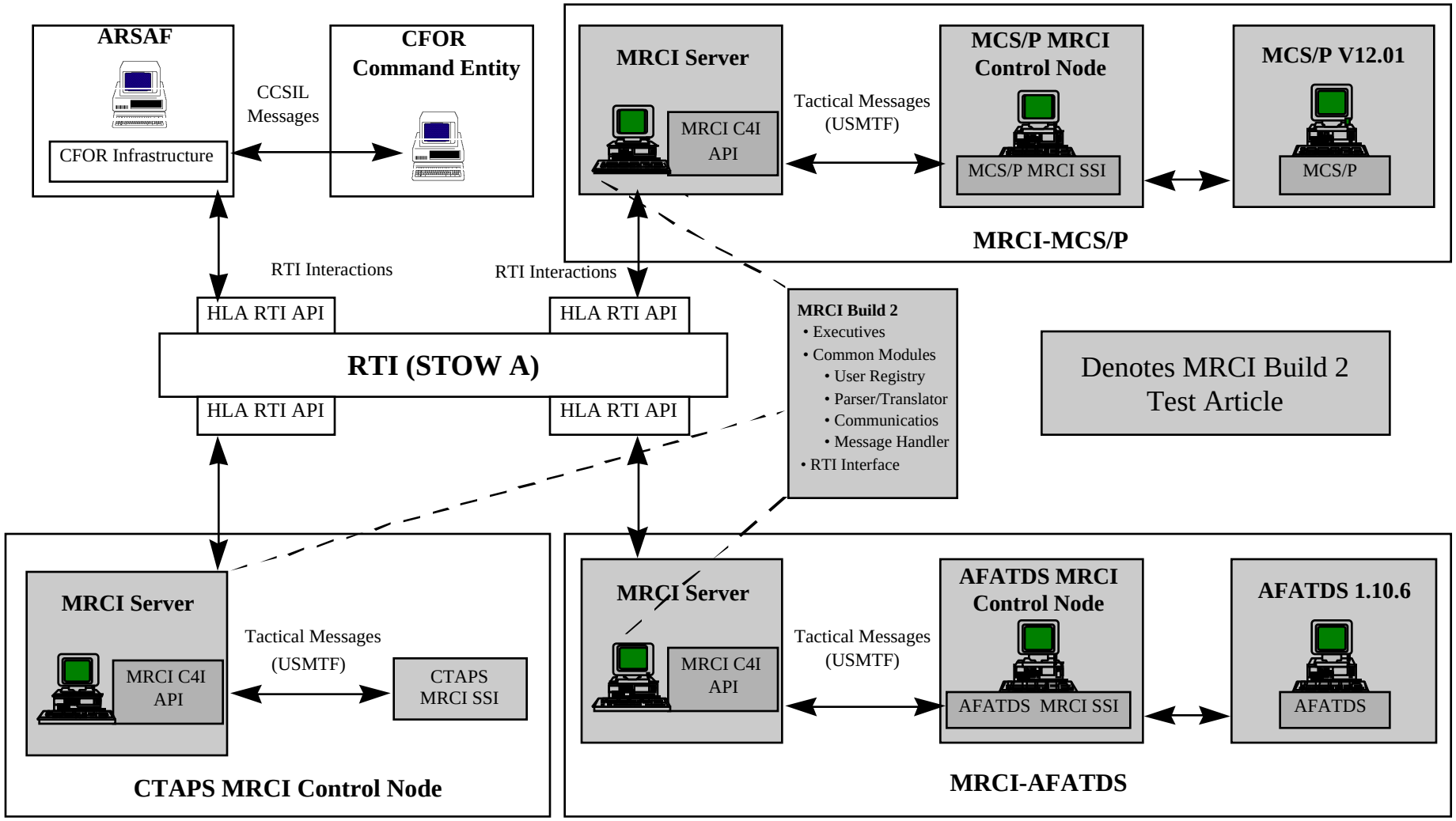
Simulation Federate
CBS

- Developer: Jet Propulsion Laboratory (JPL)
- Sponsoring Government Agency: U.S. Army Simulation, Training, and Instrumentation Command (STRICOM)
- Brief Description of Simulation: Corps Battle Simulation (CBS) provides computer-based battle simulation support for military training exercises. CBS is the ground model for the Aggregate Level Simulation Protocol (ALSP) Confederation.
- For Use In: ALSP Confederation Test 97 (CT97), Prairie Warrior 97 (PW97), Ulchi Focus Lens 97 (UFL97)
- C4I Systems Interfaced To: MCS/P & AFATDS
- Status: Currently participating in the All Actor Integration (AAI) Test at the Joint Training, Analysis, and Simulation Center (JTASC)

TRR Agenda (2 of 3)

Time	Subject	
1000-1020	MCS/P Update	- Howard
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1020-1040	AFATDS Update	- Anglin
	- SSI Implementation	
	- Mission Threads / Messages	- Griggs
1040-1100	Simulation Federate Update	- Hieb
1100-1120	Test Program	- Chen
	- CT-5	
	- Post February MRCI Assessment Opportunities	
	- CBS JTC Update	

MRCI Server, MCS/P SSI, CTAPS, and AFATDS SSI System Testing at MRCI DOC



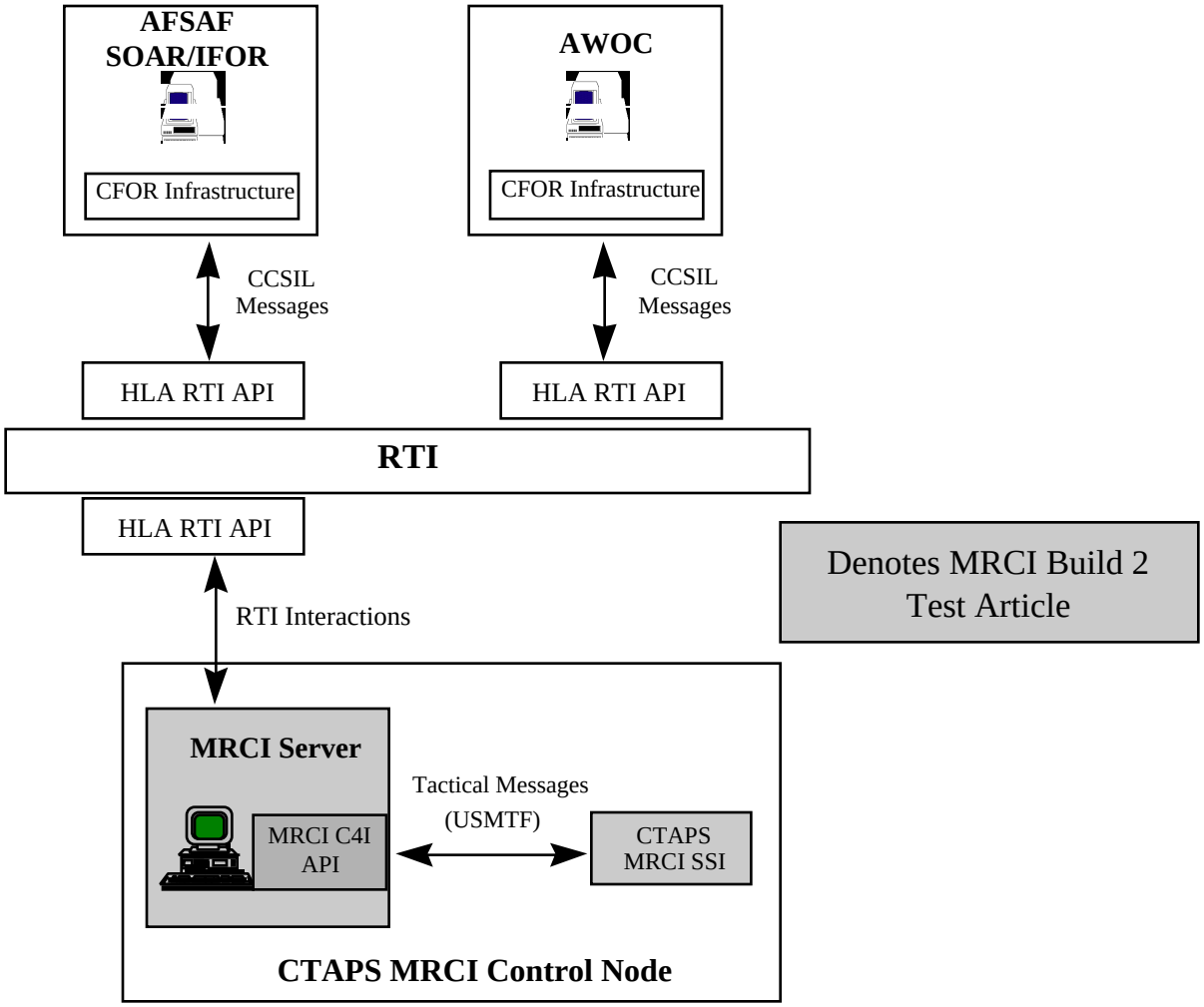
MRCI DOC Testing

Test Objectives: Demonstrate the use of MRCI software to interface to the MCS/P 12.01, AFATDS 1.10.6, CTAPS 5.1.3.

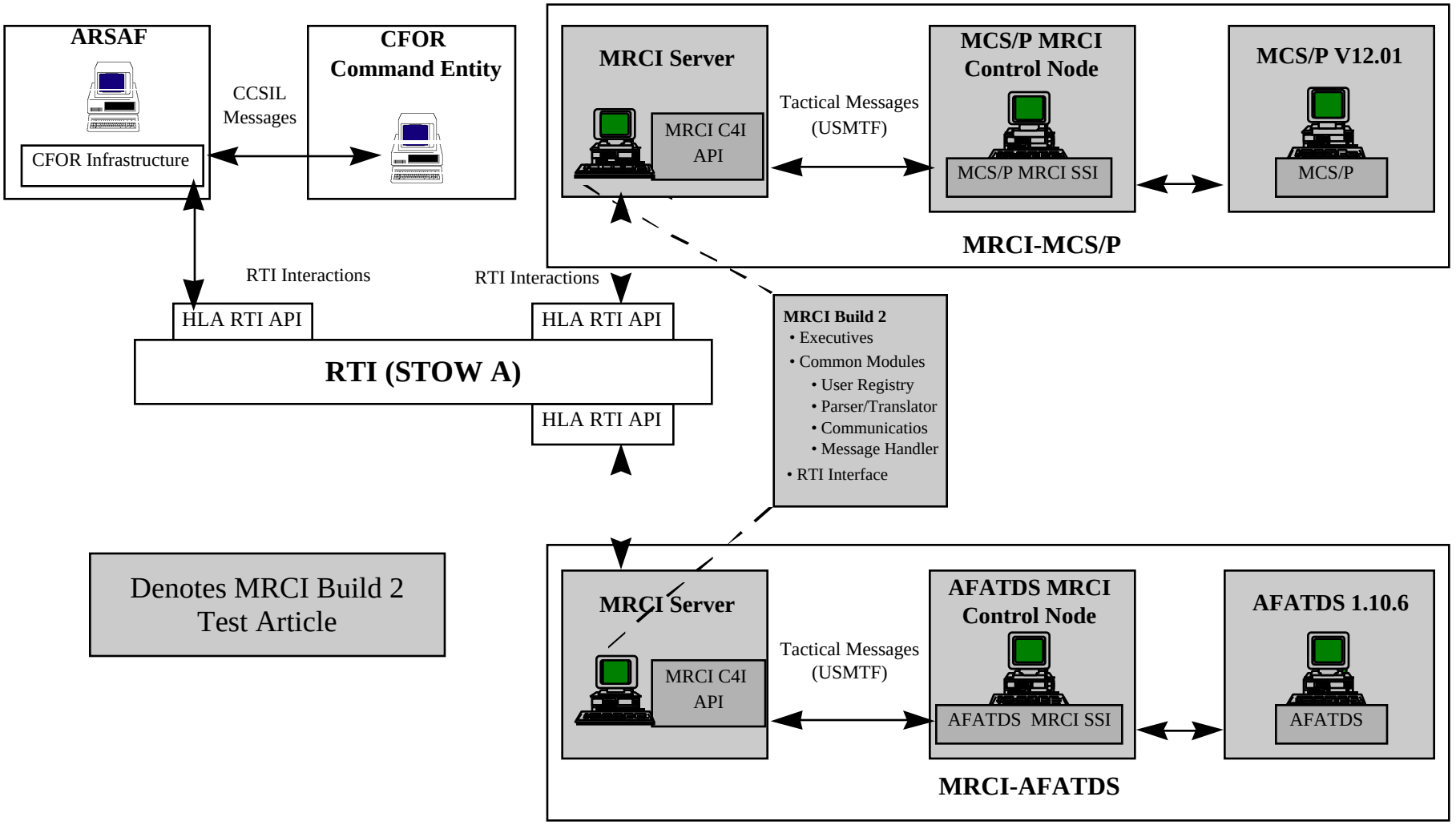
Requirement(s) to be tested:

- Verify the capability of MRCI to receive the tactical messages; to translate portions of this message into a CCSIL message; and to transmit the CCSIL message to the HLA/RTI.
- Verify the capability of the MRCI Server to receive the CCSIL messages from the High Level Architecture/Runtime Infrastructure (HLA/RTI); to translate portions of the messages into the appropriate USMTF tactical messages and to transmit those tactical messages to the MCS/P 12.01and AFATDS 1.10.6.
- Demonstrate the ability of MRCI Server to log tactical and CCSIL messages.

MRCI Build 2 Test Articles at STOW CT-5
Air Force Experiment at WISSARD, Oceana, VA



MRCI Build 2 Test Articles at STOW CT-5
Army Experiment at NSC, Leavenworth, KS



STOW CT5 Testing

Test Objective: Demonstrate the use of MRCI software to interface to the MCS/P 12.01, CTAPS V5.1.3, and the AFATDS V1.10.6.

Requirement(s) to be tested:

- Verify the capability of MRCI to receive the tactical messages; to translate portions of this message into a CCSIL message; and to transmit the CCSIL message to the HLA/RTI.
- Verify the capability of the MRCI Server to receive the CCSIL messages from the High Level Architecture/Runtime Infrastructure (HLA/RTI); to translate portions of the messages into the appropriate tactical messages and to transmit those tactical messages to the MCS/P and AFATDS.
- Verify the capability of MRCI Server to log tactical and CCSIL messages.

Data Recording and Analysis

Will be accomplished through use of a trouble reporting system. Each DTR record will include :

- Trouble Report ID
- Category/Priority
- Cooperated C4I Applications/Systems
- Problem Description
- Date Opened/Opened By
- Date Closed/Closed By
- Action Taken/Updates
- Recommendation Actions (i.e.: workaround, solution, etc.)
- Comments

Tactical Message-to-CCSIL Message Translations

Tactical Message	*CCSIL Message*
USMTF	
C400 Situation Report	#201 (Unit-Situation-Report
ATOCONF	#1500 (Air-Tasking-Order)
ACO	#1501 (Airspace-Control-Order)
ATCCS	
S201 Battlefield Geometry	#101 (Operation-Order) Operation Overlay Annex Engineer Annex
S302 Free Text Message	#103 (Execute Directive)
S309 Enemy Interoperability	#201 (Unit Situation Report)
S507L Resource Msg-Location	#201 (Unit-Situation-Report)
TACFIRE	
FM;CFF	#401 (Fire-Request)
FOCMD	#402 (Fire Mission Information and Control)
FM;SUBS	#402 (Fire Mission Information and Control)
MTO	#402 (Fire Mission Information and Control)

CCSIL Message-to-Tactical Message Translations

CCSIL Message	*Tactical Message*
#101 (Operation-Order) Operation Overlay Annex Engineer Annex	S201 Battlefield Geometry
#201 (Unit-Situation-Report)	S507L Resource Msg-Location
#201 (Unit Situation Report)	S309 Enemy Interoperability
#401 (Fire-Request)	FR GRID
#402 (Fire Mission Information and Control)	FM;FOCMD
#402 (Fire Mission Information and Control)	EOM&SURV
#1700 (Mission-Status-Report (ETD,ETA))	TACREP
#1701 (Mission-Status-Report (ATD,ATA))	TACREP
#1702 (Mission-Deviation-Report)	TACREP
#1707 (Air-Mission-Report)	MISREP

MRCI ASSESMENT SCHEDULE

MRCI Description Index ID	Description	ALSP	ASTC	JTF	STOW 97
1	MRCI execution should be transparent to the user and non-intrusive to the C4I system during setup and use.	2/24/97	12/13/96	4/6/97	12/16/96
2	MRCI shall be able to operate in real time and/or at a speed which results in the perception of real time (perceptible real time) to the C4I system using the MRCI. MRCI must not preclude or inhibit the use of time management schemes supported by the RTI.		2/13/97		
3	MRCI shall operate with dynamic changes in C4I systems task organization and in all mission threads (e.g. planning through BDA and re-planning to re-tasking).				
4	MRCI shall operate during, and recover from, system failures on either its RTI or live C4I side.		2/13/97		
5	MRCI shall support C4I systems representing echelons above Corps to platform level, e.g. infantryman operating autonomously.		2/13/97		
6	MRCI shall not restrict the HLA Federation operations with respect to security level.		2/13/97		
7	MRCI operation shall not be constrained by data, information or C2 formats and shall not introduce additional layers of complexity to the simulation interfaces to the RTI.		2/13/97		
8	MRCI shall be able to go to war and operate across operational war-fighting networks.		2/13/97		
9	MRCI shall support bi-directional interactions between C4I systems and the HLA-based Federation.	2/24/97	12/13/96	4/6/97	12/16/96
10	MRCI shall comply with the five Federation and five Federate rules of the HLA.	2/24/97	12/13/96	4/6/97	12/16/96
10.1	Federations must have an HLA Federation Object Model (FOM), documented using the HLA OMT.	2/24/97	12/13/96	4/6/97	12/16/96
10.2	In a federation, all object representation (ownership or reflection) occurs in the federates, not in the runtime infrastructure (RTI).	2/24/97	12/13/96	4/6/97	12/16/96
10.3	During a federation execution, data exchange (attribute values and interactions) among instances of objects defined in the FOM represented (owned or reflected) in different federates occurs via the RTI).	2/24/97	12/13/96	4/6/97	12/16/96
10.4	During a federation execution, federates must interact with the runtime infrastructure (RTI) in accordance with the HLA interface specification.	2/24/97	12/13/96	4/6/97	12/16/96
10.5	During a federation execution, an attribute of an instance of an object can be owned by only one federate at any given time.	2/24/97	12/13/96	4/6/97	12/16/96

MRCI ASSESMENT SCHEDULE

MRCI Requirement Index ID	Description	ALSP	ASTC	JTF	STOW 97
10.6	Federates must have an HLA Simulation Object Model (SOM) documented using the HLA OMT.	2/24/97	12/13/96	4/6/97	12/16/96
10.7	Federates must be able to publish/reflect any attributes of objects in their SOM and exercise SOM object interactions externally.	2/24/97	12/13/96	4/6/97	12/16/96
10.8	Federates must be able to own or reflect attributes and to transfer/accept ownership of attributes dynamically during a federation execution, as specified in their SOM.	2/24/97	12/13/96	4/6/97	12/16/96
10.9	Federates must be able to vary the conditions (e.g. thresholds) under which they provide updates of public attributes of objects according to their SOM.	2/24/97	12/13/96	4/6/97	12/16/96
10.10	Federates must be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation in accordance with at least one HLA time management service.	2/24/97	12/13/96	4/6/97	12/16/96
11	MRCI must facilitate inter-operation with an HLA federation using all five RTI service categories. I.e. Federation Management, Time Management, Object Management, Ownership Management and Declaration Management.	2/24/97	12/13/96	4/6/97	12/16/96
12	MRCI shall provide the throughput and transport capabilities to facilitate the rapid exchange and synchronization of C4I and Simulation databases (database reconciliation as executed by the future HLA exercise generation components.	2/24/97	12/13/96	4/6/97	12/16/96
13	MRCI shall facilitate the collection of both FOM and non-FOM data as defined within the C4I system SOM.	2/24/97	12/13/96	4/6/97	12/16/96
14	MRCI shall facilitate the establishment of an application-to-application session between the RTI and the C4I system.	2/24/97	12/13/96	4/6/97	12/16/96
15	MRCI shall provide a mechanism for re-synchronization with C4I systems following degraded operations (e.g. tactical picture reestablishment).	2/24/97	12/13/96	4/6/97	12/16/96
16	MRCI shall be GCCS DII COE compliant.				
17	MRCI applications shall be fully inter-operable with Ada 95.				
18	MRCI shall support next generation releases of C4I system software (e.g. MCS/P Baseline Build V, Block III; AFATDS V 1.10.06).	2/24/97	2/13/97	4/6/97	12/16/96
19	The MRCI/C4I SOM shall support FOMs produced for STOW demonstrations and exercises which include CBS, OpenSAF, EADSIM participation and entity-level interactions.				
20	To the extent practical, MRCI re-configurable modules shall be reusable among instances of C4I-MRCI combinations.				
21	MRCI shall support flow of both perceived and ground-truth data, information and C2 transactions consistent with applicable FOM and SOM definitions for Federations in which it participates.				

MRCI ASSESMENT SCHEDULE

MRCI Requirement Index ID	Description	ALSP	ASTC	JTF	STOW 97
22	MRCI design shall not be restricted by the use of legacy simulation-to-real world interface solutions.				
23	MRCI design shall not be restricted by the use of alternate redundant mechanisms to the RTI.				
24	MRCI shall be developed using a language for specification of formats, timing and conversion requirements of data, information and C2 interchange in clear, consistent and concise interface specifications of internal and external interfaces.	2/24/97	12/13/96	4/6/97	12/16/96
25	MRCI shall use well-defined application program interface between layers and the support services.	2/24/97	12/13/96	4/6/97	12/16/96
26	MRCI shall optimize the interdependencies between software components so that the impact of change is localized.				
27	MRCI shall reduce the number of, and special training required for, system administrators, network administrators, and other exercise support personnel.				
28	MRCI shall minimize life-cycle costs and be logistically supportable.				
29	MRCI shall be flexible, extensible, and modifiable to capitalize on current and emerging industry accepted standards and commercially available products to the maximum extent possible to support the system requirements and to streamline upgrades.	2/24/97	12/13/96	4/6/97	12/16/96
30	MRCI shall provide sufficient flexibility, modifiability and performance to support changes and extensions to the interfaces on both the C4I and RTI sides.	2/24/97	12/13/96	4/6/97	12/16/96
31	MRCI shall execute in a distributed manner across heterogeneous platforms including current war-fighting systems.				
32	MRCI software shall be portable to different vendor host platforms with minimal or no modifications.				
33	MRCI shall provide an experimental capability to interface AWSIM/R to TBMCS IAW the TBMCS SOM.				
33.1	MRCI shall provide the capability of the current PRW and CWIC interfaces.				
33.2	MRCI shall provide the capability to interface existing simulations with current and rapidly-prototyped C4I systems.				
34	MRCI shall provide an experimental capability to interface NASM/AP to TBMCS.				
34.1	MRCI shall provide the capability to be used with next generation simulations and the Prototype Federation products.				
35	MRCI shall provide an experimental capability to interface AFSAF to TBMCS.				
35.1	MRCI shall support the parsing and transmission of ATO/ACO for virtual mission planning and execution within AFSAF.				

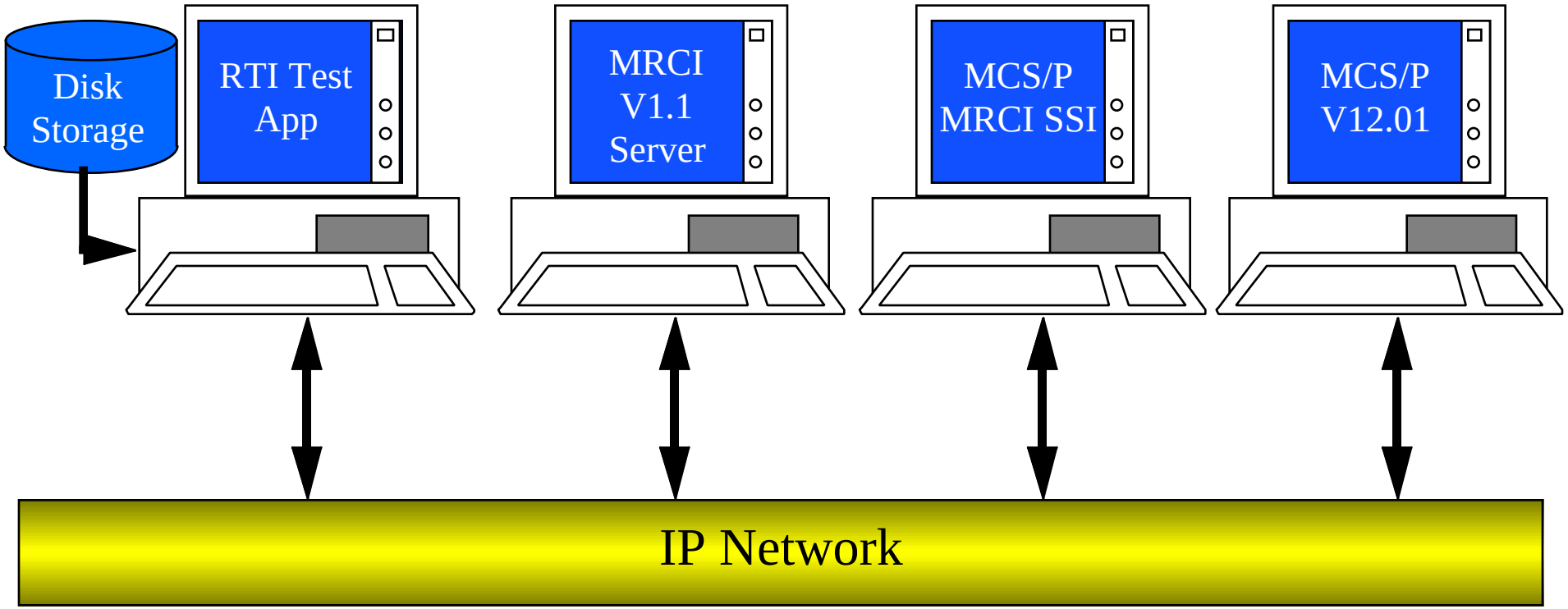
MRCI ASSESMENT SCHEDULE

MRCI Requirement Index ID	Description	ALSP	ASTC	JTF	STOW 97
35.2	MRCI shall support operations in Federations where STOW SEID SI and OpenSAF are used IAW the appropriate FOM.				
36	The design of the MRCI shall not preclude the addition of a module to support direct C4I system database access (vice message interchange) when specified in future C4I SOMs.				
37	MRCI must support segregation, labeling and simultaneous existence of live and simulation data within all of its modules and in all of its outputs on both C4I and RTI sides.				
38	MRCI must support the populating of messages with relatively unstructured text content to the C4I system and within the CCSIL-like message converter, while correctly maintaining the intent of such messages.				
39	MRCI must support interpreting messages with relatively unstructured text content from the C4I system and within the CCSIL-like message converter, while correctly maintaining the intent of such messages.				
40	The Federation Design in which the MRCI participates must accommodate scaling, normalizing or otherwise harmonizing data and information transactions where “detail mismatches” would result in unrealistic representations of the battlespace to the C4I system.				
41	MRCI must provide functionality compatible with the STOW SSF and data collection facilities in support of STOW FOMs.				
42	MRCI must maintain content integrity and conformity in all internal data-to-data/information-to-information/C2-to-C2 transformations.				
43	MRCI must not introduce spatial or temporal inconsistencies into the C4I system’s “world view”.				
43.1	Via the MRCI, simulated entities must be able to affect the live C4I systems and vice versa. Simulated entities must also be able to control communications between live C4I systems; data, information, and C2 flow between live and simulated world shall be influenced in quantity and quality based on environment, geometric, physics and other connectivity determinants computed elsewhere in the Federation.				

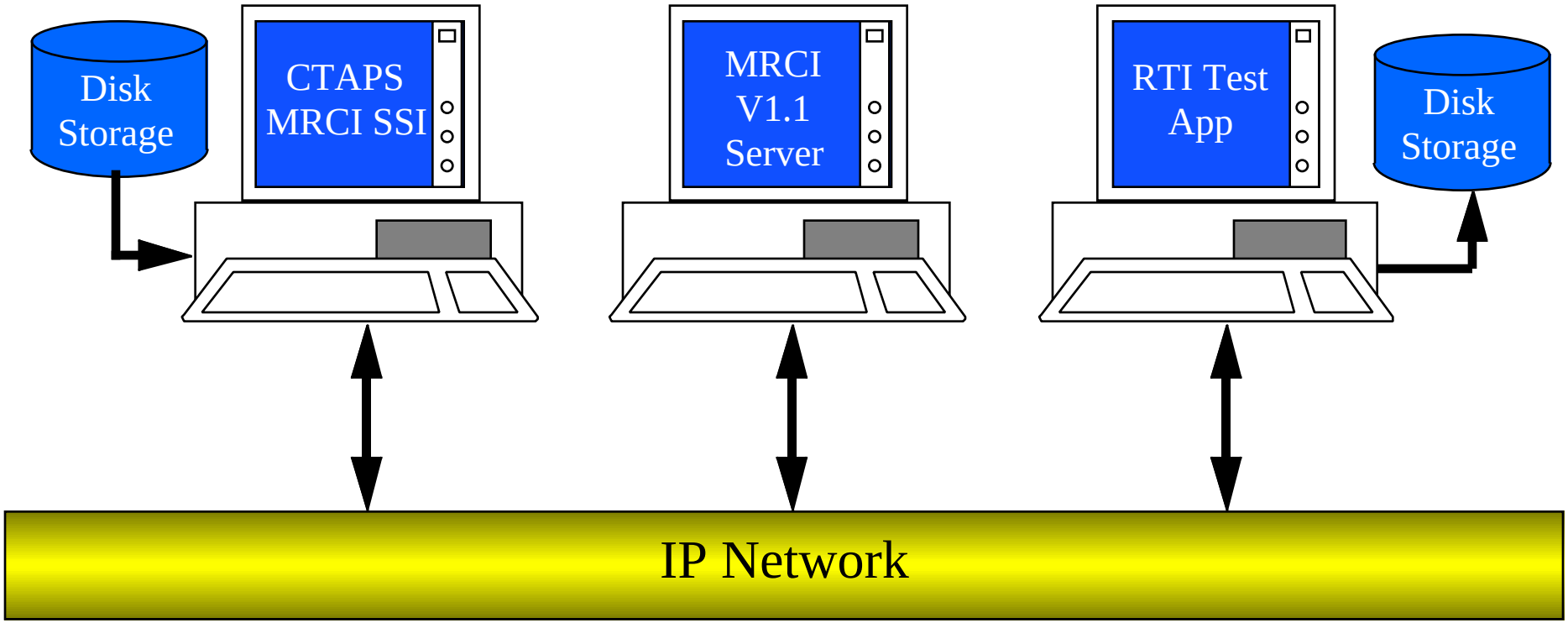
TRR Agenda (3 of 3)

Time	Subject	
1120-1150	**MRCI Demonstrations**	**- Griggs/ Hieb**
1150-1200	**Wrap-up**	**- Park**
1200	**Adjourn**	

MCS/P Demonstration Setup



CTAPS Demonstration Setup



TRR Agenda (3 of 3)

Time	Subject	
1120-1150	MRCI Demonstrations	- Griggs/ Hieb
1150-1200	Wrap-up	- Park
1200	Adjourn	

Deliveries Since PTRR

- **Test Procedures (A001BB-2/BD-2), CTAPS-AFSAF, MCS/P & AFATDS-ARSAF/CFOR**
- **Software Users Manual, Programmer Notebook - Draft**
  - **A007AA SOW 3.1, MRCI Framework (S/W Reqmts Spec)**
  - **A007AB SOW 3.2, MRCI SSI - CTAPS/MCS/P, AFATDS**
  - **A007AC SOW 3.3, MRCI CMs - CTAPS/MCS/P, AFATDS**
  - **A007AD SOW 3.4, MRCI RTII**
  - **A007AE SOW 3.5, Integrate MRCI Components**
- **SOMs:**

CTAPS	**11 Oct 96 (Draft)**	**13 Jan 97 (Update)**
MCS/P	**11 Oct 96 (Draft)**	**13 Jan 97 (Update)**
AFATDS	**11 Oct 96 (Draft)**	**13 Jan 97 (Update)**

TRR Agenda (3 of 3)

Time	Subject	
1120-1150	MRCI Demonstrations	- Griggs/ Hieb
1150-1200	Wrap-up	- Park
1200	**Adjourn**	